Naval Research Laboratory

Washington, DC 20375-5000

NRL Report 9301

High-Quality 800-b/s Voice Processing Algorithm

G. S. KANG AND L. J. FRANSEN

Human-Computer Interaction Lab
Information Technology Division

February 25, 1991

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 25, 1991	Interim

4. TITLE AND SUBTITLE

High-Quality 800-b/s Voice Processing Algorithm

5. FUNDING NUMBERS

61153N
X7290-CC
DN280-290

6. AUTHOR(S)

G. S. Kang and L. J. Fransen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL Report 9301

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Space and Naval Warfare Systems Command
Arlington, VA 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The 2400-b/s linear predictive coder (LPC) is currently being widely deployed to support tactical voice communication over narrowband channels. However, there is a need for lower-data-rate voice encoders for special applications: improved performance in high-bit-error conditions, low-probability-of-intercept (LPI) voice communications, and integrated voice/data systems.

As a result of continued research the intelligibility of very-low-data-rate (600 to 800 b/s) voice processors has steadily improved, from the upper 70s about a decade ago to the middle 80s in recent years. This report presents a new 800-b/s voice-encoding method that produces an intelligibility score of 92 (measured by the Diagnostic Rhyme Test (DRT)). This high score compares favorably with that attained by the 2400-b/s LPC.

14. SUBJECT TERMS

Speech analysis synthesis
Low-bit-rate speech encoding
Matrix quantization of spectral parameters
Speech parameter extraction
Vector quantization of amplitude

15. NUMBER OF PAGES
42

16 PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

CONTENTS

HIGH-QUALITY 800-b/s VOICE PROCESSING ALGORITHM

1. INTRODUCTION

The linear predictive coder (LPC) operating at 2400 bits per second (b/s) is being widely deployed to support tactical voice communication over narrowband (approximately 3 kHz) channels. One example of the LPC is the Advanced Narrowband Digital Voice Terminal (ANDVT or AN/USC-43(V)) for tri-service tactical application. According to the latest estimate, the Navy is procuring 11,900 ANDVTs. Another example is the Subscriber Terminal Unit — third generation (STU-III) used by the civilian sector of the Government. All told, a large number of 2400-b/s LPCs will be in operation, and they will be in service well into the next century.

Recently, however, voice processors operating at much lower data rates than 2400 b/s (i.e., 600 to 800 b/s) have been sought for various specialized applications:

- Increased tolerance to bit errors — The intelligibility of the 2400-b/s LPC degrades rather quickly in the presence of bit errors. With 3% random errors, the intelligibility decreases to below 79, a level often described as having "poor intelligibility." To increase the tolerance to bit errors, error protection code is added to the very-low-data--rate (600 to 800 b/s) speech for transmission at 2400 b/s. Some years ago, we studied this approach [1]. We have been told that this approach is currently being considered for implementation in the United States and abroad. We are providing the 800-b/s voice algorithm for this effort.

- Low probability of intercept (LPI) — If the speech data rate is lower, we can transmit speech over channels having a smaller bandwidth and/or shorter time interval. Thus, an indispensable element of an LPI voice system is a voice processor operating at very low data rates. A great deal of effort is in progress to implement LPI voice terminals.

- Narrowband voice/data integration — Recently, voice/data integration has drawn much attention. If the channel capacity is 2400 b/s, digital data can be transmitted simultaneously with voice data by removing perceptually insignificant bits from the 2400-b/s LPC bit stream and replacing them with digital data. We investigated this method a few years ago [2]. According to our experiments, digital data up to 80 b/s can be transmitted simultaneously with voice data without degrading speech intelligibility or causing operational incompatibility with other 2400-b/s LPCs that do not have this capability. If we encode speech below 2400 b/s, however, we can transmit more digital data simultaneously with voice. Currently, the Navy is developing a narrowband voice/data integration capability through the Shared Adaptive Inter-Networking Technology (SAINT) program. We are contributing voice algorithms for this effort.

Manuscript approved October 1, 1990.

In this report we describe an 800-b/s voice processor for these applications. The intelligibility of this voice processor is 92 as measured by the Diagnostic Rhyme Test (DRT) for the reference condition (i.e., noise-free speech using three male speakers). This is the highest score achieved by an 800-b/s voice processor to this date under the same reference condition. This result compares favorably with the 2400-b/s LPC of just a few years ago.

We wrote this report for three groups of people: program managers and sponsors who are actively involved in the transfer of voice technology to working hardware; communication-architecture planners who are interested in the state of the art of voice encoders; and independent researchers who develop voice processors. We hope that this report provides some useful information to these individuals.

2. BACKGROUND

In this report, we chose 800 b/s as the data rate for encoding speech because this is the lowest data rate at which we can achieve "very good" intelligibility, as shown in Fig. 1. A data rate of 800 b/s is not a standard transmission rate (i.e., 75^n b/s, n = 1, 2, ...). For the three applications previously mentioned, however, the 800-b/s voice data will be supplemented with other data prior to transmission. Therefore, the output data rate will be a standard rate.

For the past 10 years we have been investigating voice encoders operating at 800 b/s (Fig. 1). Speech intelligibility has increased almost 10 points during this time. Since a data rate of 800 b/s is approximately 1% of the data rate associated with unprocessed speech, some degradation of speech is inevitable. But some of our early 800-b/s voice processors were rather unintelligible. Once, we played the game "battleship" over a two-way link by using a real-time 800-b/s voice processor (1984 version listed in Fig. 1). The speech intelligibility was so low that some listeners could not discriminate between a hit or a miss.

Some low-data-rate voice processors are still inferior. Recently (May 1, 1990), we read about a 600-b/s voice processor that achieved a DRT score of only 76.0. Many critical factors must be carefully examined to achieve an acceptable voice quality at these low data rates. We discuss these critical factors in succeeding sections.

Low-data-rate voice processors (operating at data rates of 2400 b/s or below) use a simple electric analog of the human voice system to synthesize speech (Fig. 2). The speech model shown in Fig. 2(b) can be controlled by as few as 13 parameters. Despite its simplicity, the model is capable of providing adequate communicability, particularly for experienced tactical communicators.

The all-pole filter is the most frequently used vocal tract filter. According to our tests, the all-pole filter is the most efficient and reliable form of the vocal tract filter because the poles are dependent only on past input speech samples. Pole-zero vocal tract filters have been mentioned in the past. According to our experimentation, however, the inclusion of a few zeros in the vocal tract filter has not markedly improved speech intelligibility or quality. Furthermore, estimation of zeros are not that reliable because the zeros are dependent on the estimated past output samples; thus, estimated output errors tend to significantly affect the subsequent zero estimation.

In the past, the excitation signal for low-data-rate voice processors has been either a pulse train (to generate voiced speech) or random noise (to generate unvoiced speech). Recently, spectrally shaped random noise has been added to the voiced excitation signal, and spikes have been superimposed on the unvoiced excitation signal at speech onset [11]. The addition of random noise in the voiced excitation signal produces sustained vowels that sound less "buzzy" because the speech waveform does not repeat

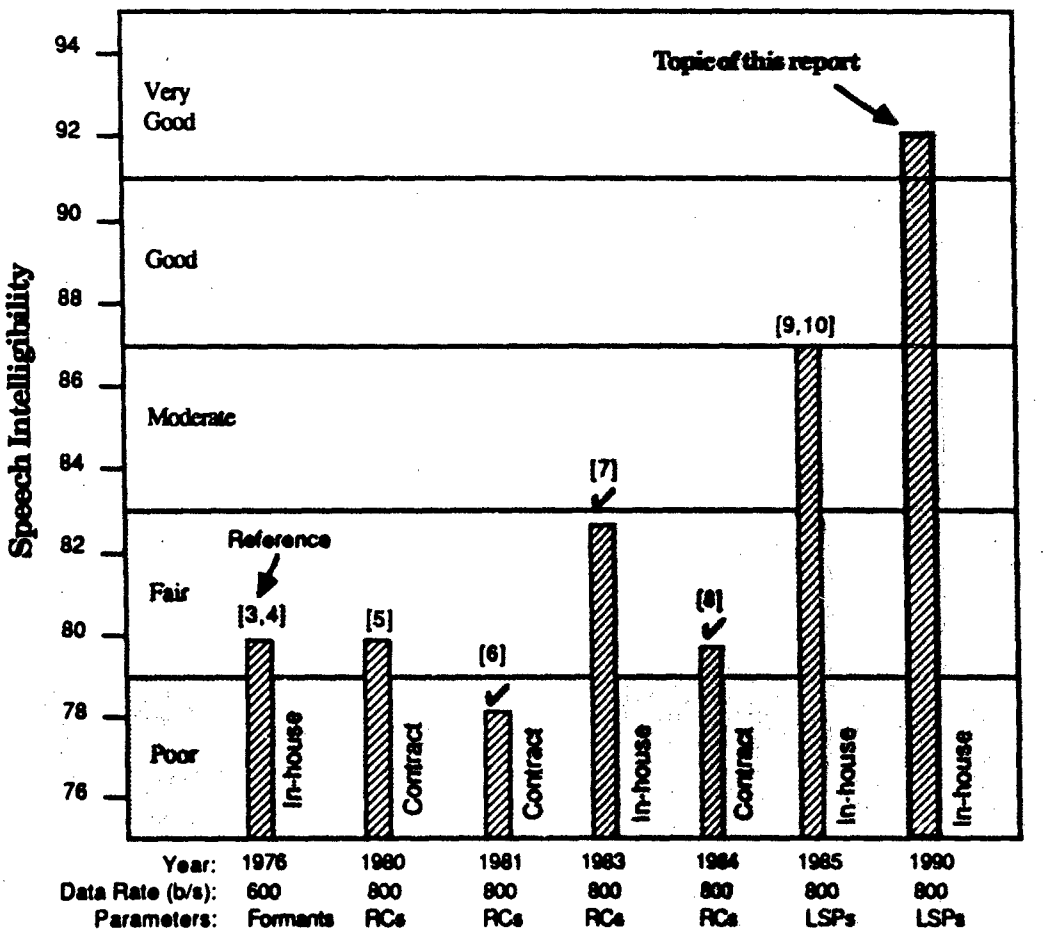
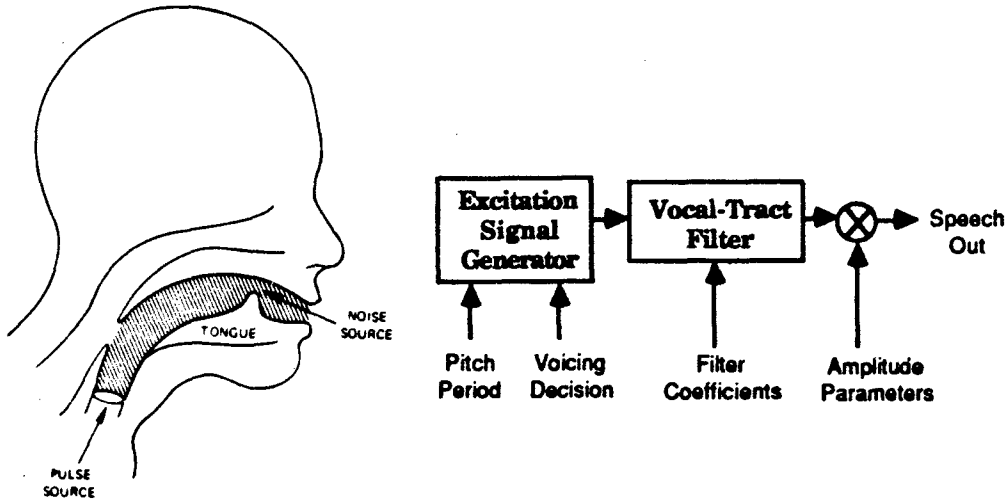


Fig. 1 — Low-data-rate voice processors developed at the Naval Research Laboratory. Real-time processors are identified by (√). As shown, intelligibility of 800-b/s encoded speech has steadily improved over the years [3-10]. The most striking difference between the two most recent processors and the others is the use of speech parameters called "line spectrum pairs (LSPs)" rather than reflection coefficients (RCs) used in the 2400-b/s LPC. The descriptors "very good," "good," "fair," etc. have been adopted by the DoD Digital Voice Processor Consortium.



(a) Human speech system (b) Electric analog of (a)

Fig. 2 — Human speech production system and a simple electrical analog used to generate 800-b/s speech. All the speech parameters except the pitch period are updated approximately 50 times a second.

exactly from one pitch period to the next (as in natural speech). On the other hand, the addition of spikes in the unvoiced excitation (only at the onset of stop consonants) produces stop consonants that are appropriately abrupt. Years ago, "cat" often sounded like "hat" because of a lack of spikiness in the unvoiced excitation at the onset of stop consonants. This is no longer the case.

3. TECHNICAL APPROACH

The simple speech model shown in Fig. 2(b) has been successfully implemented at a data rate of 2400 b/s. For experienced communicators, it is an acceptable system. The 2400-b/s system updates all the parameters at each frame. We followed this basic principle in our 800-b/s voice processor. Thus, none of the speech parameters are encoded differentially in our 800-b/s voice processor; therefore, an error in one frame will not affect subsequent frames. Our approach is summarized as follows:

- Pitch period — The pitch resolution is typically 20 steps per octave over three octaves. We reduced it to 12 steps per octave over a pitch range slightly less than three octaves (i.e., pitch period from 20 to 120 sampling time intervals). Thus, the pitch is encoded to a five-bit quantity (i.e., 32 possible combinations). Furthermore, we transmit the pitch period once every other frame because the pitch contour does not change radically during normal conversation. The pitch resolution is coarser than that of the 2400-b/s LPC, but it is not discernible to casual listeners. Note that the entire five bits are transmitted every other frame.

- Amplitude parameter — The amplitude resolution of a 2400-b/s LPC is typically 1.875 dB per step over a 60 dB dynamic range (i.e., a five-bit quantity or 32 possibilities). By jointly (or vectorially) encoding amplitude parameters from two adjacent frames, we achieved a 10-bit amplitude resolution over two frames by using only nine bits. A saving of one bit per two frames is realized by excluding improbable amplitude transitions from one frame to the next. Certain amplitude transitions (viz., a 60 dB loudness variation in

20 ms) are improbable because our lungs and vocal tract cannot produce such an extreme loudness change in such a short time. Note that each amplitude index is associated with two amplitude values, one each from two adjacent frames. Thus, in effect, we transmit one amplitude value in each frame, similar to the 2400-b/s LPC.

- Filter coefficients — The 10 filter coefficients for the 2400-b/s LPC are quantized individually into 41 bits (i.e., 21.2 trillion spectral combinations). These filter parameters are capable of reproducing speech as well as nonspeech sounds. We can reduce the number of bits to encode filter parameters through a pattern-matching technique (i.e., vector quantization) in which the reference templates contain filter coefficients generated by only human voices. Furthermore, if we jointly encode filter coefficients of two consecutive frames, we not only eliminate filter coefficients capable of producing nonspeech sounds from the coding table, but we also eliminate improbable filter coefficient transitions associated with normal speech. We used this two-dimensional vector quantization (called matrix quantization) in our 800-b/s voice processor. Note that we transmit two LSP vectors in two frames.

- Voicing decision — In general, voiced speech spectra and unvoiced speech spectra are recognizably different. For example, no voiced speech spectra are without the first formant frequency. For the first time, we have embedded the voicing decision with the filter coefficients.

Figure 3 is a block diagram of our 800-b/s voice encoder. As noted, a number of blocks are also used in the 2400-b/s LPC, but they are not discussed in this report. The blocks unique to 800-b/s voice encoding are discussed in the remainder of this report.

4. CRITICAL FACTORS

Frame Size

Frame size is the time interval between parameter updates. In the past, frame size was often determined after considering the number of bits required to encode all the parameters per frame. This is not a good design approach because there is a preferred value for frame size in terms of speech intelligibility for voice processors that use an artificial excitation signal (i.e., pitch-excited vocoders such as the 2400 LPC and the 800-b/s voice processor). In these voice processors, rapid speech changes can be reproduced only by rapid filter and amplitude parameter updates. Intelligibility is adversely affected by slow speech onsets.

Contrary to the conventional design practice, we fixed the frame rate first, based on the highest speech intelligibility attainable for the pitch-excited vocoder, then computed the number of bits necessary to encode speech parameters at 800 b/s. There are many ways to encode speech parameters efficiently, but speech degradation resulting from improper frame size is irreversible.

Some years ago, a study was conducted to investigate the relationship between frame size and speech intelligibility [13]. According to this study, a marked speech degradation occurs as the frame size increases from 20 to 30 ms. Recently, we also examined the effect of frame size on speech intelligibility as measured by the DRT. By using a 10-tap LPC without parameter quantization, we obtained DRT scores for three frame sizes: 17.5 ms, 20 ms, and 22.5 ms (Fig. 4). (As indicated in Fig. 4, a frame of 20 ms is the preferred choice.)

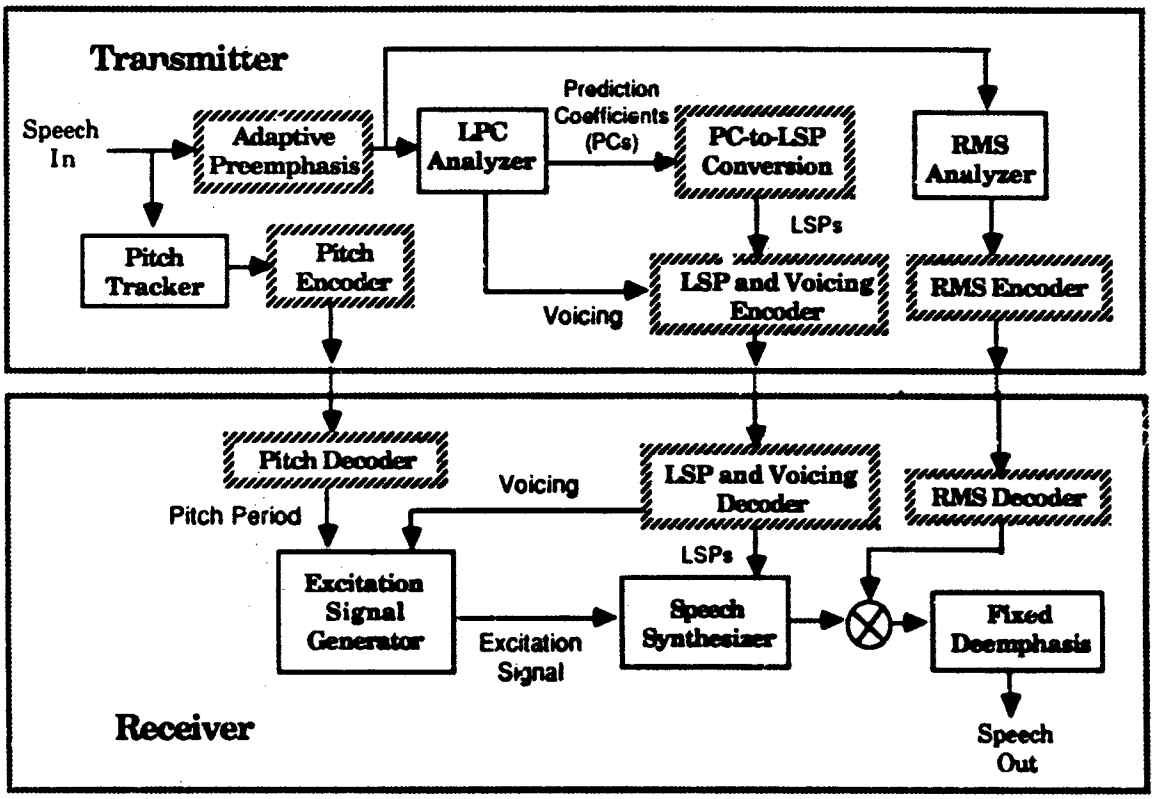


Fig. 3 — Block diagram of our 800-b/s LPC. The thin-lined boxes are also used in the 2400-b/s LPC [12]. Therefore, they are not discussed in this report. The hatched boxes are unique to the 800-b/s voice processor, and they are discussed in subsequent sections.

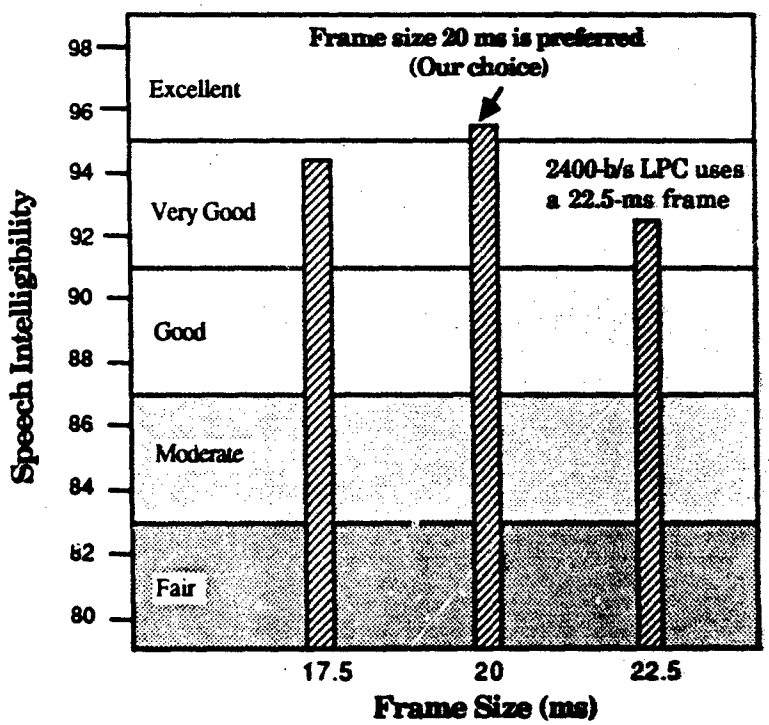


Fig. 4 — Frame size vs speech intelligibility. This figure shows DRT scores for a 10-tap LPC with three different frame sizes. Most 2400-b/s voice processors have a frame size of 22.5 ms, but the preferred size is 20 ms (which is used in our 800-b/s voice processor). It is significant that a pitch-excited LPC that uses an artificial excitation signal (i.e., a pulse train for voiced speech or random noise for unvoiced speech) can achieve a DRT score of 95 with unquantized parameters.

Number of Filter Coefficients

The number of filter coefficients is also a critical factor for the pitch-excited vocoder because the spectral envelope of synthesized speech is determined solely by the filter coefficients. The choice of an optimum number of filter coefficients, however, is not as straightforward as the choice of an optimum frame size because the choice is directly related to the pitch period of the speech waveform. For example, 16 coefficients provide higher speech quality than 10 coefficients for low-pitch male voices (see Fig. 5) because they approximate the speech spectral envelope more faithfully; they produce more focused speech sounds, particularly for sustained vowels.

On the other hand, 16 coefficients will generate reverberant speech for high-pitch female voices because 16 coefficients tend to characterize sparsely spaced pitch harmonics rather than the spectral envelope (see Fig. 6). It is significant to note that the LPC spectrum does not approximate the speech spectral envelope of a female voice as well as that of a male voice. This is because the speech waveform has more pitch epochs per frame, and the principle of linear prediction does not hold well near the pitch epoch where the ongoing speech waveform is disturbed by the glottis excitation.

In terms of intelligibility, however, the number of coefficients (between 10 and 16) is not too sensitive for both male and female voices. A larger number of coefficients improves the spectra of sustained vowels rather than the fast-changing speech onsets that affect the DRT scores. We think that 10 coefficients is an adequate choice for the 800-b/s voice processor.

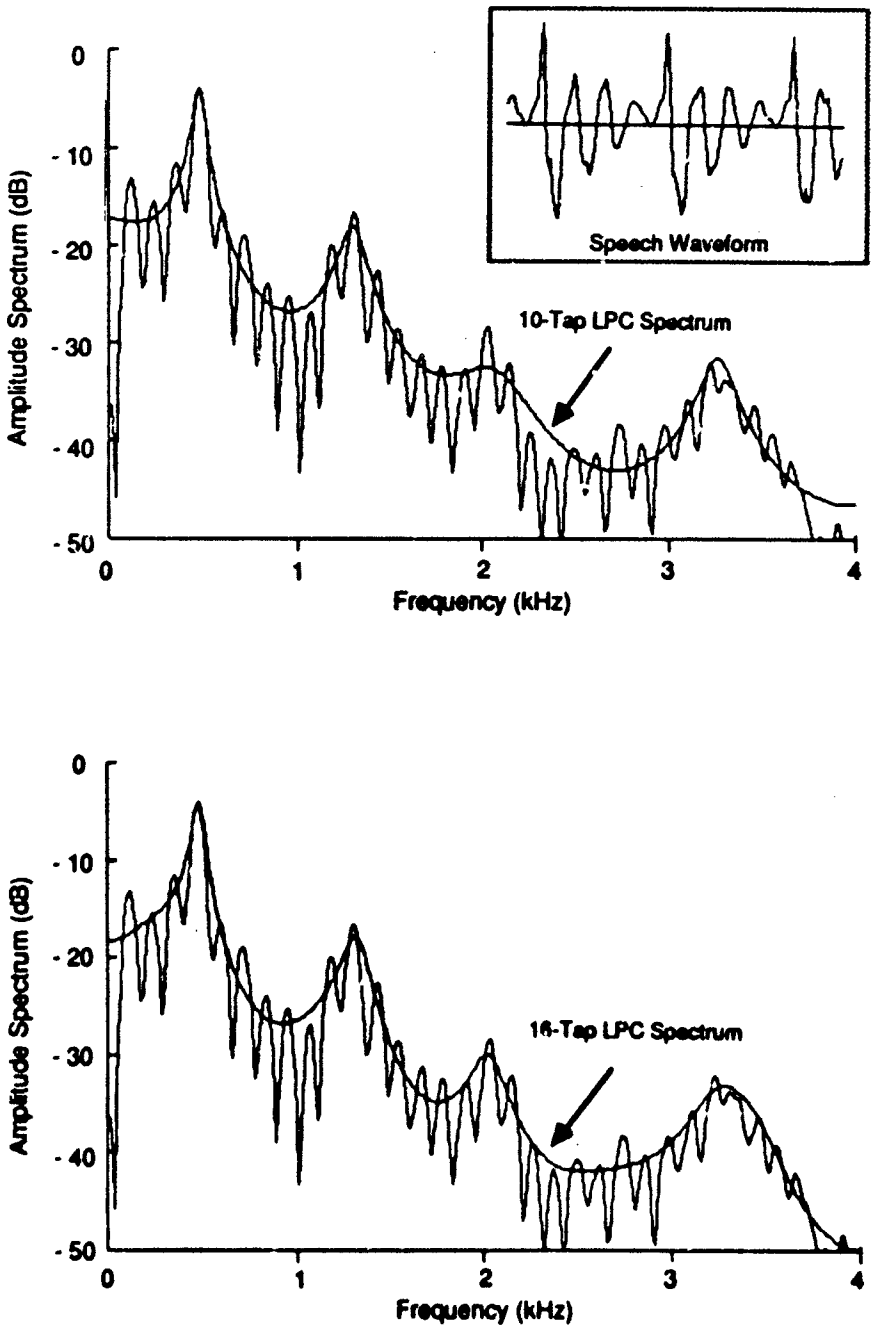


Fig. 5 — Male voice speech spectrum with superimposed LPC spectrum taken from the sustained vowel in the word /show/. As noted, the LPC spectrum approximates the speech spectral envelope more accurately when the number of coefficients is increased from 10 to 16. Pitch harmonics of a low-pitch male voice are closely spaced, as shown in this figure; thus, the LPC spectrum cannot follow pitch harmonics (which is good).

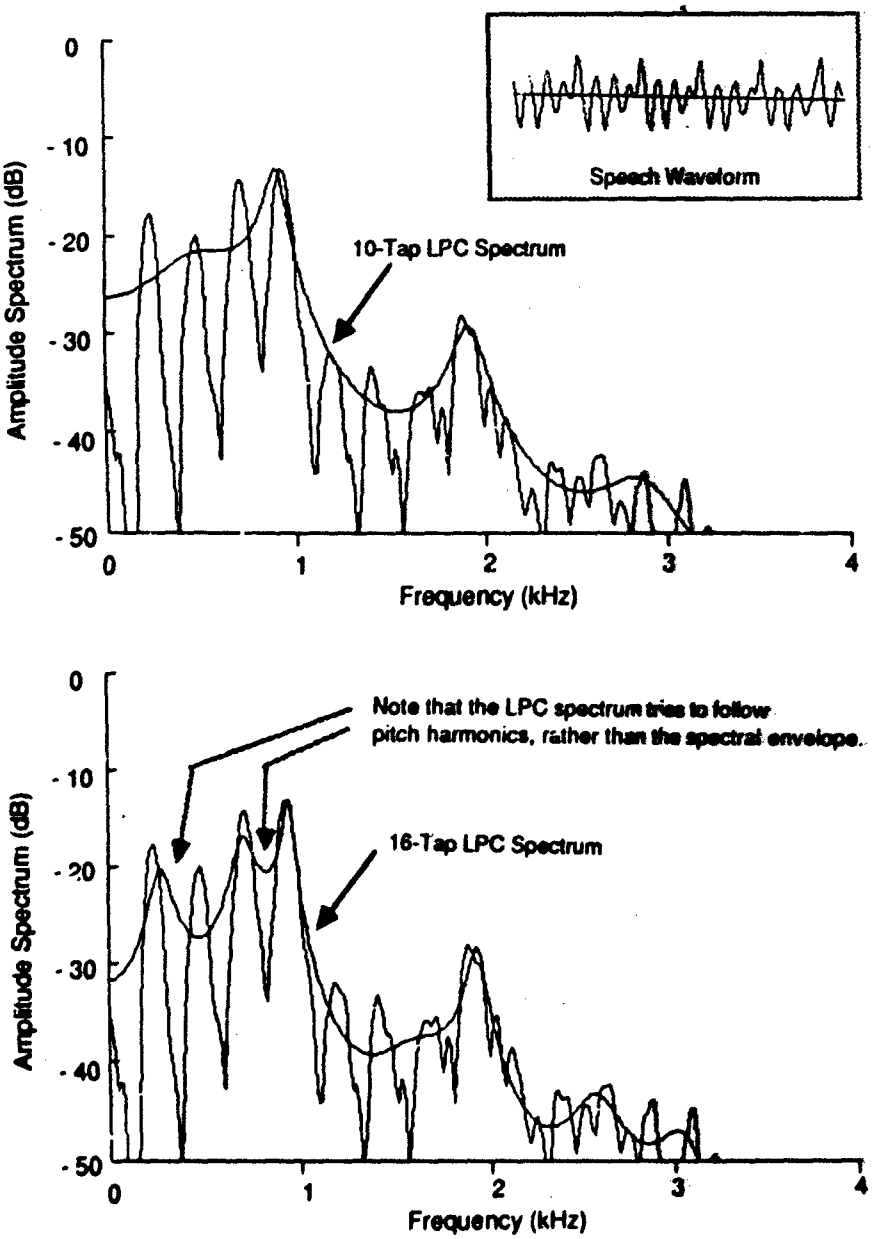
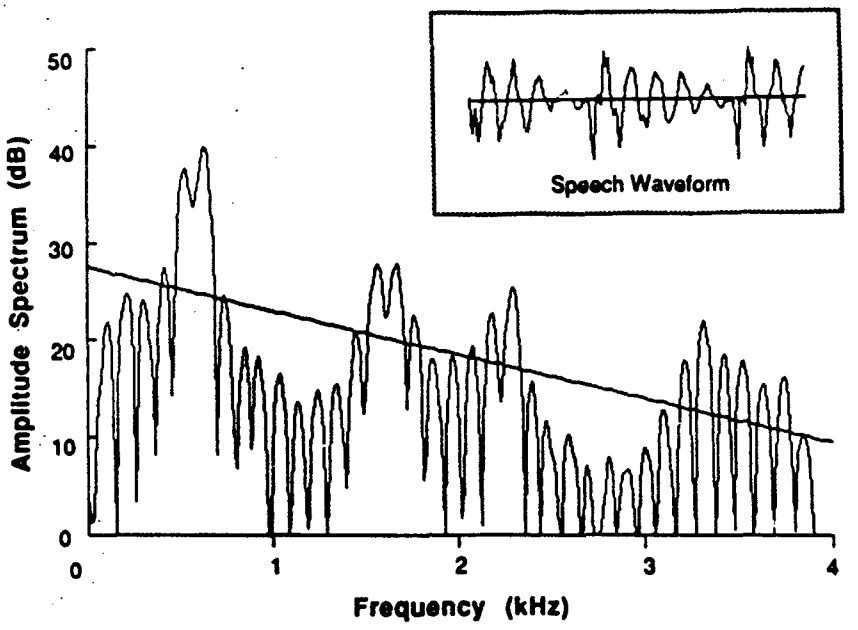


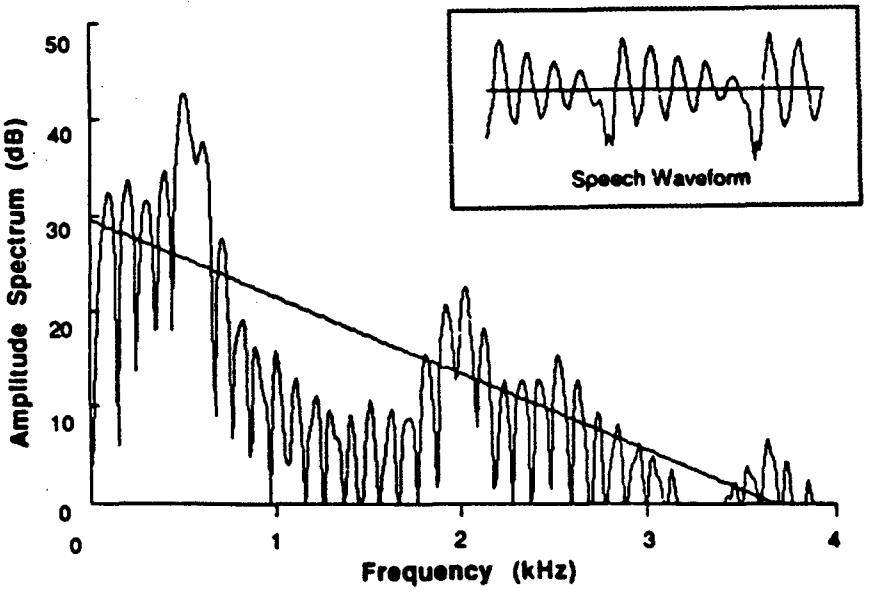
Fig. 6 — Female voice speech spectrum with superimposed LPC spectrum taken from the sustained vowel in the word /yes/. The 16-tap LPC spectrum tends to follow pitch harmonics rather than the spectral envelope. The resultant speech is reverberant. For female voices, 8 to 12 coefficients are adequate.

Spectral Tilt Equalization (Adaptive Preemphasis)

A clear ringing voice has more high-frequency energies (Fig. 7(a)) because of favorable glottis and vocal tract characteristics; these include: glottis closes instantly (i.e., wideband excitation); glottis closes completely (i.e., good "on-and-off" contrast); and vocal tract is not lossy (i.e., no speech leakage from the nasal passages). On the other hand, certain voices have weak upper bands (Fig. 7(b)) because their glottis and vocal characteristics do not produce high-frequency energies.



(a) Clear ringing voice



(b) Somewhat "muddy" voice

Fig. 7 — Speech spectra of the vowel /a/ in "way" from two different persons. Figure 7(a) is an example of a clear and ringing voice that is not easily drowned by ambient noise (good voice for cocktail parties). Figure 7(b) represents a typical aging voice that lacks high-frequency energies. The LPC analysis disfavors the speech spectrum that is heavily tilted. Thus, LPC analysis is usually preceded by preemphasis (high-frequency boosting), often using a single-zero filter, $1 - (31/32)z^{-1}$.

We know that LPC analysis does not work as well for speech signals having weak upper-frequency components. Therefore, LPC analysis is often preceded by preemphasis (high-frequency boost). Usually, a fixed preemphasis is used. Since the magnitude of the spectral tilt varies from person to person, adaptive preemphasis is preferred in which the amount of high-frequency boost is controlled by the amount of spectral tilt of the input speech.

Adaptive preemphasis is accomplished by a single-zero filter with an adaptive filter weight:

$$y(i) = x(i) - \beta\, x(i-1) \tag{1}$$

where β is the adaptive-preemphasis factor, and $x(i)$ and $y(i)$ are the input and output speech samples. We chose β to be the coefficient of the first-order linear predictor because it approximates the speech envelope by a single variable, and this variable contains mainly information regarding the spectral tilt. Thus,

$$\beta = \frac{E[x(i)\, x(i-1)]}{0.5\{E[x^2(i)] + E[x^2(i-1)]\}}, \tag{2}$$

where E[.] signifies the running average of the past history when using a single-pole low-pass filter. The feedback gain of the low-pass filter is a critical factor. We recommend a feedback gain somewhere between 0.990 and 0.995, which is large enough for the output be more dependent on the speaker's vocal timber than speech itself.

The theoretical range of β in Eq. (2) is -1.0 to 1.0. If the speech signal generates β values around 0.5 or less, the speech waveform already has strong high-frequency components (i.e., unvoiced fricatives /s/, /sh/, /ch/, etc.); hence, no further preemphasis is needed. Therefore, we let 0.5 be the minimum value of β for the preemphasis operation defined by Eq. (1).

The purpose of adaptive preemphasis is to reduce the variability of the spectral tilt from one voice to another. Thus, adaptive preemphasis is expected to produce a fewer number of unique spectral templates for a given population size. As a result, each spectral template will represent a speech sound from a greater number of people. To verify this hypothesis, we collected spectral templates (detailed procedures are discussed later) from five sentences each from 54 males and 12 females. The total number of spectral patterns with a fixed preemphasis was 37,172, whereas the total number of spectral patterns with an adaptive preemphasis was 34,032 (8.4% reduction). This is a sizable reduction in template sizes. Significantly, speech intelligibility is not degraded by adaptive preemphasis.

Lastly, the adaptive preemphasis factor (β) is not transmitted. In essence, the adaptive preemphasizer is a signal conditioner at the front-end of the voice processor. At the receiver, fixed deemphasis (with a deemphasis factor of 0.75), similar to the conventional 2400-b/s LPC, is used.

LSPs as Filter Parameters

As noted in Fig. 1, the intelligibility of 800-b/s voice processors improves significantly after LSPs are used as filter parameters. LSPs have been gaining interest because their intrinsic properties permit more efficient encoding than the better-known reflection coefficients (RCs):

- Frequency-selective spectral error — An error in one member of the LSPs affects the spectrum only near that frequency (i.e., frequency selective). Thus, LSPs may be

quantized in accordance with properties of auditory perception (i.e., coarser representation of the higher-frequency components of the speech-spectral envelope).

- Unequal spectral-error sensitivity — For a given LSP set, spectral-error sensitivity of each line spectrum can be determined easily (as will be shown). Thus, fewer bits are needed to encode spectrally less sensitive LSPs.

We have presented various aspects of LSPs in an NRL report [9]. In this section we present essential aspects of LSPs beneficial to low-bit-rate speech encoding.

Computational Procedures

LSPs are obtained by transforming the prediction coefficients generated by the linear predictive analysis. In linear predictive analysis, a speech sample is represented as a linear combination of past samples. Thus,

$$x_i = \sum_{k=1}^{10} \alpha(k)\, x_{i-k} + \epsilon_i \,, \tag{3}$$

where x_i is the ith speech sample, $\alpha(k)$ is the kth prediction coefficient (PC), and ϵ_i is the ith error (prediction residual) sample. The LPC analysis filter, $A(z)$, that transforms speech samples to residual samples is expressed by

$$A(z) = 1 - \sum_{k=1}^{10} \alpha(k)\, z^{-k} \qquad \text{[LPC Analysis Filter]} \tag{4}$$

where z^{-1} is a one-sample delay operator.

$A(z)$ may be decomposed to a set of two transfer functions, one having an even symmetry and the other having an odd symmetry. This can be accomplished by taking a difference and sum between $A(z)$ and its conjugate function $A^*(z)$ (i.e., the transfer function of the filter whose impulse response is a mirror image of $A(z)$). Thus,

$$P(z) = A(z) + z^{-11} A^*(z) \qquad \text{[Sum Filter]} \tag{5}$$

and

$$Q(z) = A(z) - z^{-11} A^*(z) \qquad \text{[Difference Filter]} \,. \tag{6}$$

Table 1 lists the coefficients of both sum and difference filters.

The impulse response of the sum filter has an even symmetry with respect to its midpoint (see Table 1 or Fig. 8). The filter has six roots along the unit circle, as indicated by small squares in the z-plane shown in Fig. 8. A real root located at 4 kHz is extraneous. The frequencies corresponding to these roots are upper LSP frequencies.

Table 1 — Coefficients of Sum and Difference Filters, *P(z)* and *Q(z)*, for the 10th-order LPC Analysis Filter

Sum Filter	Difference Filter
P(1) = 1.	Q(1) = 1.
P(2) = - [PC(1) + PC(10)]	Q(2) = - [PC(1) - PC(10)]
P(3) = - [PC(2) + PC(9)]	Q(3) = - [PC(2) - PC(9)]
P(4) = - [PC(3) + PC(8)]	Q(4) = - [PC(3) - PC(8)]
P(5) = - [PC(4) + PC(7)]	Q(5) = - [PC(4) - PC(7)]
P(6) = - [PC(5) + PC(6)]	Q(6) = - [PC(5) - PC(6)]
P(7) = - [PC(6) + PC(5)] = P(6)	Q(7) = - [PC(6) - PC(5)] = - Q(6)
P(8) = - [PC(7) + PC(4)] = P(5)	Q(8) = - [PC(7) - PC(4)] = - Q(5)
P(9) = - [PC(8) + PC(3)] = P(4)	Q(9) = - [PC(8) - PC(3)] = - Q(4)
P(10) = - [PC(9) + PC(2)] = P(3)	Q(10) = - [PC(9) - PC(2)] = - Q(3)
P(11) = - [PC(10) + PC(1)] = P(2)	Q(11) = - [PC(10) - PC(1)] = - Q(2)
P(12) = 1. = P(1)	Q(12) = - 1. = - Q(1)

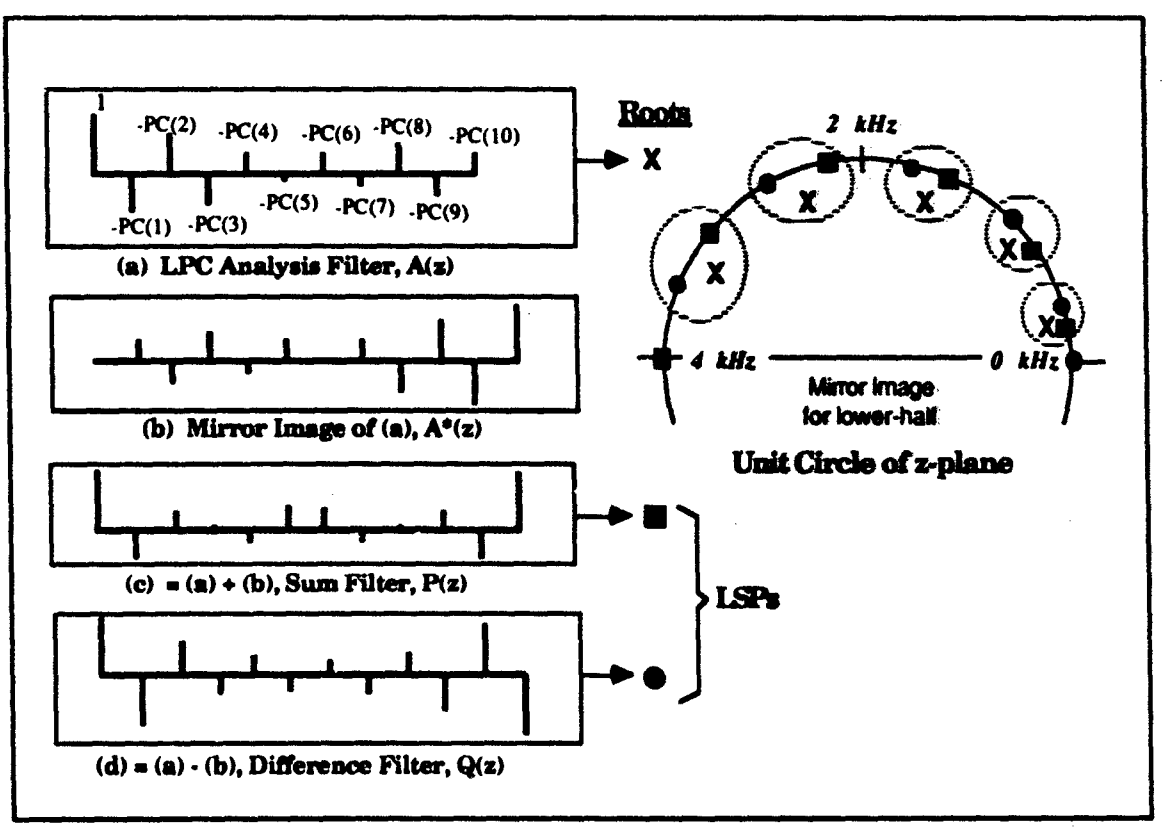


Fig. 8 — Decomposition of LPC analysis-filter impulse response. The LPC analysis filter shown in Fig. 8(a) is replaced by the sum and difference filters shown in Figs. 8(c) and 8(d). No information is lost through this decomposition because Fig. 8(a) can be reconstructed from Figs. 8(c) and 8(d). An advantage of using the sum-and-difference filters is that their roots are located along the unit circle of the complex z-plane. Thus, root finding needs a one-dimensional search.

The impulse response of the difference filter has an odd symmetry with respect to its midpoint (see Table 1 or Fig. 8). The filter also has six roots along the unit circle, as indicated by small circles in the z-plane shown in Fig. 8. A real root at 0 Hz is extraneous. The frequencies corresponding to these roots are lower LSP frequencies.

The LPC analysis filter, reconstructed by the use of these two filters, is

$$A(z) = (1/2)[P(z) + Q(z)] \qquad \text{[LPC Analysis Filter]} \tag{7}$$

in which the roots of $P(z)$ and $Q(z)$ are LSPs. The amount of computation required to convert the PCs to LSPs is substantial. Any root-finding technique that relies on convergence of the solution is not recommended for real-time voice encoding because it is difficult to estimate the computation time since the number of iterations to obtain a solution varies significantly from one coefficient set to another.

In the past various methods of converting from prediction coefficients (PCs) to LSPs have been studied. One interesting example is the use of Chebyshev polynomials [14]. We also developed an algorithm for converting PCs to LSPs. The algorithm requires a fixed amount of computation for each conversion. The algorithm has been implemented for real-time operation by using Texas Instruments' TMS320C25 fixed-point microprocessor and, more recently by using TMS320C30 floating-point microprocessor and the SKYBOLT (INTEL i860) acceleration board.

PC-to-LSP Conversion

LSPs are null frequencies associated with the frequency responses of sum and difference filters, $P(z)$ and $Q(z)$. The null frequencies are obtained by local minima of the frequency responses as the frequency is scanned from 0 to 4 kHz at a 20 Hz step. Each null frequency is refined through a parabolic interpolation by using three consecutive spectral points.

To reduce computations, we first remove the extraneous roots at $z = 1$ and $z = -1$. They are time-invariant, and they contain no speech information that can be factored out. Then both sum and difference filters have even-symmetric impulse responses. Real-root removed sum and difference filters are obtained from

$$P(z) = (1 + z^{-1})PP(z) \tag{8}$$

and

$$Q(z) = (1 - z^{-1})QQ(z). \tag{9}$$

The coefficients for $PP(z)$ and $QQ(z)$ are obtained by polynomial division. Table 2 lists the results. As noted in the table, the impulse responses of the real-root removed $P(z)$ or $Q(z)$ are even symmetric, and only six values are unique.*

*Even symmetry of PP(z) given in Table 2 may be proven by the following steps:

PP(7) = P(7) - PP(6)
= P(6) - PP(6) [See Table 1 for P(7) = P(6)]
= P(6) - [P(6) - PP(5)] [See Table 2 for PP(6) = P(6) - PP(5)]
= PP(5)

PP(8) = PP(4), or QQ(8) = QQ(4), etc, can be proven by a similar procedure.

Table 2 — Coefficients of Real-Root Removed Sum and Difference Filters, $PP(z)$ and $QQ(z)$

Real-Root Removed Sum Filter	Real-Root Removed Difference Filter
PP(1) = 1.	QQ(1) = 1.
PP(2) = P(2) - PP(1)	QQ(2) = Q(2) + QQ(1)
PP(3) = P(3) - PP(2)	QQ(3) = Q(3) + QQ(2)
PP(4) = P(4) - PP(3)	QQ(4) = Q(4) + QQ(3)
PP(5) = P(5) - PP(4)	QQ(5) = Q(5) + QQ(4)
PP(6) = P(6) - PP(5)	QQ(6) = Q(6) + QQ(5)
PP(7) = P(7) - PP(6) - PP(5)	QQ(7) = Q(7) + QQ(6) - QQ(5)
PP(8) = P(8) - PP(7) - PP(4)	QQ(8) = Q(8) + QQ(7) - QQ(4)
PP(9) = P(9) - PP(8) - PP(3)	QQ(9) = Q(9) + QQ(8) - QQ(3)
PP(10) = P(10) - PP(9) - PP(2)	QQ(10) = Q(10) + QQ(9) - QQ(2)
PP(11) = 1. - PP(1)	QQ(11) = 1. - QQ(1)

Since $P(z)$ and $Q(z)$ are related to prediction coefficients (see Table 1), $PP(z)$ and $QQ(z)$ can be expressed directly in terms of prediction coefficients. Table 3 lists the results.

Table 3 — Coefficients of Real-Root Removed, Sum and Difference Filters in Terms of Prediction Coefficients

Real-Root Removed Sum Filter	Real-Root Removed Difference Filter
PP(1) = 1.	QQ(1) = 1.
PP(2) = - [PC(1) + PC(10)] - PP(1)	QQ(2) = - [PC(1) - PC(10)] + QQ(1)
PP(3) = - [PC(2) + PC(9)] - PP(2)	QQ(3) = - [PC(2) - PC(9)] + QQ(2)
PP(4) = - [PC(3) + PC(8)] - PP(3)	QQ(4) = - [PC(3) - PC(8)] + QQ(3)
PP(5) = - [PC(4) + PC(7)] - PP(4)	QQ(5) = - [PC(4) - PC(7)] + QQ(4)
PP(6) = - [PC(5) + PC(6)] - PP(5)	QQ(6) = - [PC(5) - PC(6)] + QQ(5)
PP(7) = PP(5)	QQ(7) = QQ(5)
PP(8) = PP(4)	QQ(8) = QQ(4)
PP(9) = PP(3)	QQ(9) = QQ(3)
PP(10) = PP(2)	QQ(10) = QQ(2)
PP(11) = PP(1)	QQ(11) = QQ(1)

LSPs can be determined by the null frequencies of the amplitude responses of (real-root removed) sum and difference filters. A direct Fourier transform (not FFT) can be used for computing the spectra based on the first six time samples listed in Table 3. A frequency step of 20 Hz is adequate.

The amplitude response of the (real-root removed) sum or difference filter is obtained by a direct Fourier transform of the filter impulse response. The spectra of $PP(z)$ and $QQ(z)$ are computed at a 20 Hz interval from 0 to 4000 Hz. To simplify notations, let $\beta = (\pi/4000)(20)$. The amplitude response of $PP(z)$, denoted by $PP(k)$, can be obtained from

$$PP(k) = \left\{\sum_{j=1}^{11} PP(j)\cos[\beta(k-1)(j-1)]\right\}^2 + \left\{\sum_{j=1}^{11} PP(j)\sin[\beta(k-1)(j-1)]\right\}^2 \quad k = 1,2,\ldots,200 \tag{10}$$

where k is the frequency index ($k = 1$ means 0 Hz, $k = 2$ means 20 Hz, ...), and j is the time index ($j=1$ means $t = 0$ s, $j = 2$ means 125 μs, ...). Similarly, the amplitude response of $QQ(z)$, denoted by $QQ(k)$, can be expressed as

$$QQ(k) = \left\{\sum_{j=1}^{11} QQ(j)\cos[\beta(k-1)(j-1)]\right\}^2 + \left\{\sum_{j=1}^{11} QQ(j)\sin[\beta(k-1)(j-1)]\right\}^2 \quad k = 1,2,\ldots,200. \tag{11}$$

Both $PP(z)$ and $QQ(z)$ are even symmetric (see Table 3) with six unique time-samples. Thus, Eqs. (9) and (10) can be simplified to

$$PP(k) = \left\{\sum_{j=1}^{6} PP(j)\,CT(k,j)\right\}^2 + \left\{\sum_{j=1}^{6} PP(j)\,ST(k,j)\right\}^2 \quad k = 1,2,\ldots,200 \tag{12}$$

and

$$QQ(k) = \left\{\sum_{j=1}^{6} QQ(j)\,CT(k,J)\right\}^2 + \left\{\sum_{j=1}^{6} QQ(j)\,ST(k,j)\right\}^2 \quad k = 1,2,\ldots,200 \tag{13}$$

where $CT(k, j)$ and $ST(k, j)$ are cosine and sine values expressed by

$$\begin{aligned} CT(k,J) &\triangleq \cos[\beta(k-1)(j-1)] + \cos[\beta(k-1)(11-j)] \quad \text{for } j = 1,2,3,4,5 \\ &\triangleq \cos[\beta(k-1)(j-1)] \qquad\qquad\qquad\quad \text{for } j = 6. \end{aligned} \tag{14}$$

and

$$\begin{aligned} ST(k,j) &\triangleq \sin[\beta(k-1)(j-1)] + \sin[\beta(k-1)(11-j)] \text{ for } j = 1,2,3,4,5 \\ &\triangleq \sin[\beta(k-1)(j-1)] \text{ for } j = 6. \end{aligned} \tag{15}$$

The total number of cosine or sine values equals the product of the highest frequency and time indices (i.e., 200 × 6 = 1200). Among them, only 400 cosine and sine values are unique for a frequency resolution of 20 Hz and speech sampling rate of 8000 Hz. To make the implementation simpler, however, the entire 1200 cosine and sine values can be stored in sequence.

LSPs are the frequencies at which the amplitude responses of $P'P'(z)$ or $Q'Q'(z)$ vanish. To determine these frequencies, three consecutive amplitude values (A_1, A_2, and A_3) are subject to a parabolic fitting if the center value is lowest (i.e., $A_2 < A_1$ and $A_2 < A_3$). (See Fig. 2.) Let the equation of a parabola that goes through these three spectral points be expressed by

$$A(f) = af^2 + bf + c \tag{16}$$

where a, b and c are constants.

Let the coordinates of three consecutive spectral points be denoted by (1, A_1), (0, A_2), and (-1, A_3). Substituting these coordinates into Eq. (15) gives

$$\begin{aligned} A_1 &= a + b + c \\ A_2 &= c \\ A_3 &= a - b + c. \end{aligned} \tag{17}$$

From these three equations, a and b are obtained from

$$a = .5(A_3 - 2A_2 + A_1)$$

and

$$b = .5(A_1 - A_3). \tag{18}$$

At the peak or null of the parabola, the first derivative of $A(f)$ with respect to frequency must be zero. From Eq. (15), this frequency is expressed as

$$f = -b/a. \tag{19}$$

At f = f, the parabola is at the null (not the peak) because the second derivative of $A(f)$ with respect to f (i.e., $2a$) is positive because $A_2 < A_1$ and $A_2 < A_3$ in Eq. (18).

Substituting Eq. (17) into Eq. (18), the null frequency in terms of three consecutive spectral points is expressed as

$$f = .5\,(A_3 - A_1)/(A_1 - 2A_2 + A_3) \quad \text{for } A_2 < A_1 \text{ and } A_2 < A_3. \tag{20}$$

Equation (19) is the amount of normalized frequency that must be shifted with respect to the center frequency (see Fig. 9). Since one unit of normalized frequency corresponds to 20 Hz, the amount of frequency that must be shifted from the center frequency is $20f$ Hz. Thus, a line spectrum frequency is the sum of the center frequency and $20f$ Hz.

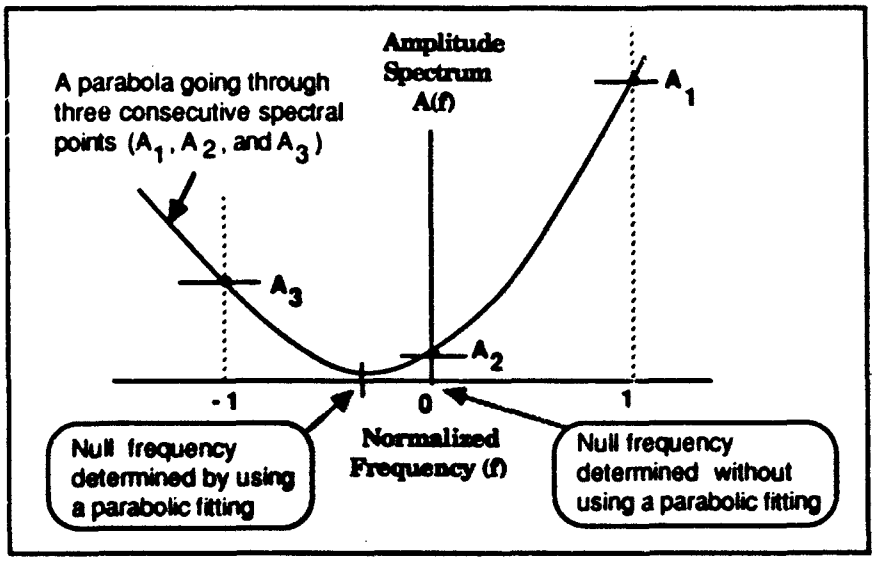


Fig. 9 — Estimation of LSPs by a parabolic fitting of three consecutive spectral values (A_1, A_2, and A_3) if $A_2 < A_1$ and $A_2 < A_3$. For convenience, the origin of the frequency axis is placed at the center frequency, and 20 Hz is normalized being unity.

LSP-to-PC Conversion

A set of LSPs can be converted to a set of PCs. The conversion algorithm can be derived in the following manner. The transfer function of the sum filter in terms of LSPs is

$$P(z) = (1 + z^{-1} \prod_{k=1}^{5} [1 - \exp(j\theta_k)\, z^{-1}][1 - \exp(-j\theta_k)\, z^{-1}] \qquad (21)$$

where θ_k is the location of the lower frequency of the kth LSP. If a line-spectrum frequency is 0 Hz, then $\theta_k = 0$ rad; if a line-spectrum frequency is 4 kHz (half sampling frequency), then $\theta_k = \pi$ rad.

Likewise, the transfer function of the difference filter is

$$Q(z) = (1 - z^{-1}) \prod_{k=1}^{5} [1 - \exp(j\theta_k')\, z^{-1}][1 - \exp(-j\,\theta_k')\, z^{-1}] \qquad (22)$$

where θ_k' is the location of the upper frequency of the kth LSP.

From Eq. (6), the transfer function of the LPC analysis filter in terms of the sum and difference filter is

$$A(z) = (1/2)[P(z) + Q(z)] \qquad (23)$$

which is in the form of

$$A(z) = 1 + \mu_1 z^{-1} + \mu_2 z^{-2} + \ldots + \mu_{10} z^{-10} \tag{24}$$

where μ's are new coefficients of $A(z)$. Comparing Eq. (3) with Eq. (22) indicates that

$$PC(k) = -\mu_k. \tag{25}$$

Typical LSP Trajectories

LSP trajectories of a spoken voice are computed by using the PC-to-LSP conversion algorithm and are plotted in Fig. 10(a). From the same speech waveform, the spectrogram is also generated and plotted in Fig. 10(b). As noted, there are similarities between them because both are frequency-domain parameters.

Hearing Sensitivity to Frequency Difference

An error in one line spectrum affects the all-pole representation of the spectrum near that frequency [9]. Thus, LSPs can be quantized according to the frequency-dependent auditory perception characteristics. For example, the ear cannot resolve differences at high frequencies as accurately as it can at low frequencies; thus, higher frequency LSPs may be quantized more coarsely than lower ones without introducing audible speech degradation.

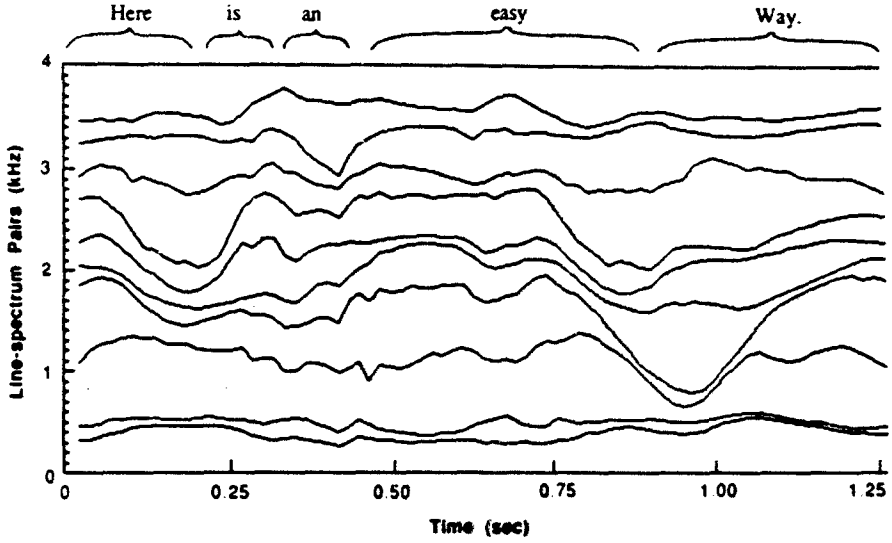
The amount of frequency variation that produces a just-noticeable difference of a single tone is approximately linear from 0.1 to 1 kHz, and it increases logarithmically from 1 to 10 kHz [15] (Fig. 11). At NRL a similar relationship was obtained for a speech-like sound by using a pitch-excited LSP speech synthesizer, with one of the 10 line spectra incrementally changed while the others remained equally spaced (i.e., resonant-free condition). This result is also shown in Fig. 11. It is expected that the curve of actual speech sounds would be located somewhere between these two curves. Figure 11 indicates that the allowable frequency difference near 4 kHz can be twice as large as that near 0 kHz.

Spectrol-Error Sensitivity of LSP

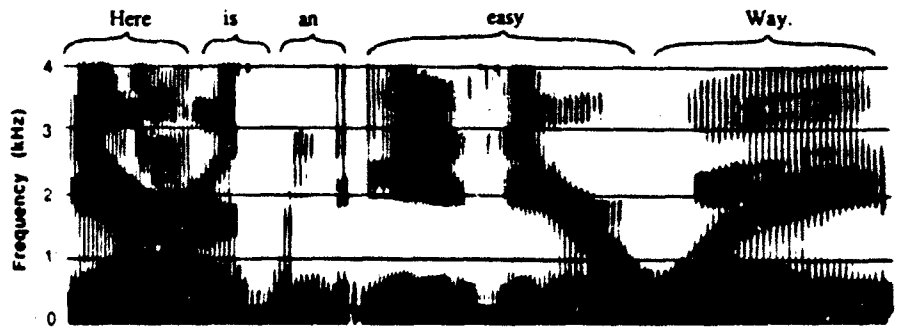
According to our observation, there is as much as a 10-to-1 difference in the spectral-error sensitivity from one line spectrum to the next. Spectrally less sensitive line-spectra should be quantized correspondingly more coarsely because they are less significant to synthesized speech.

When each line spectrum is perturbed, there is a corresponding spectral error in the frequency response of the LPC analysis filter $A(z)$ appearing in Eq. (3). The spectral-error sensitivity is a factor relating error in each line spectrum (in Hz) and the average spectral error in $A(z)$ (in dB). To derive such an expression, however, is untractable. Also, a cross-coupling of all line-spectrum errors into the overall spectral error makes the use of such an expression impractical. Therefore, a relationship that relates the average spectral error $A(z)$ to all of the line-spectrum errors (hence, including the effect of cross-couplings) is derived numerically by using various speech samples.

There is no approximation in computing the average-spectral error of $A(z)$ from given line-spectrum errors. However, to make the error expression simpler, it is necessary to impose a condition that each line spectrum have an error proportional to the frequency separation to its closest neighbor. This assumption holds well when tested with a variety of speech samples. Figure 12 is a resultant scatter plot.



(a) LSP trajectories



(b) Spectrogram

Fig. 10 — Comparison of LSP trajectories and spectrogram derived from the same speech. As noted, line-spectrum frequencies are close together where formant frequencies occur. Undistinctive and fuzzy speech often lacks closely spaced LSPs; warbling speech often has uneven LSP trajectories. We note that LSPs are effective speech parameters for diagnosing the cause of flaws in synthetic speech.

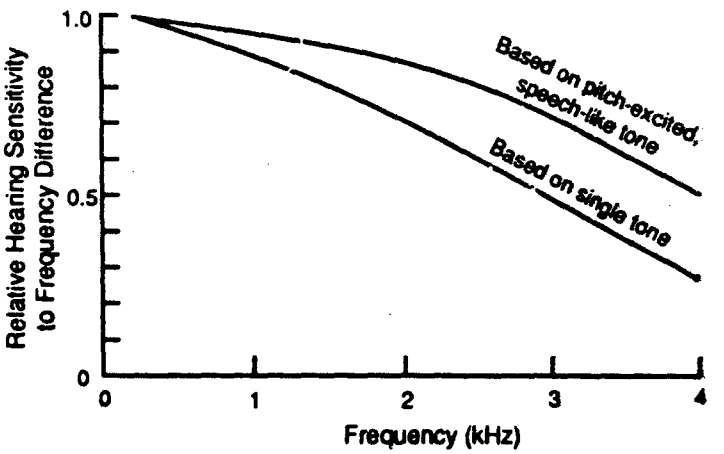


Fig. 11 — Relative hearing sensitivity to frequency difference. The result using a single tone is taken from Ladefoged [15]. The result using pitch-excited sound was taken from Kang and Fransen [9]. In both cases, relative hearing sensitivity decreased with increased frequency. This figure indicates that higher frequency LSPs need not be quantized as accurately as those of lower frequency.

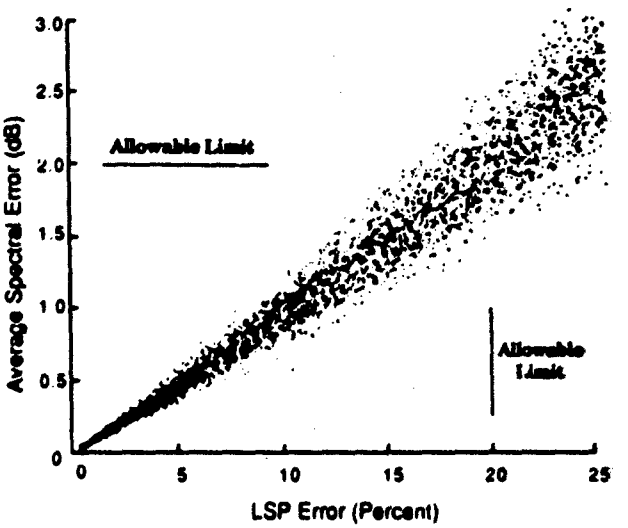


Fig. 12 — Scatter plot of average-spectral errors caused by the individual LSP errors. According to listening tests, synthesized speech is free from flutter if the average-spectral error is limited to approximately 2 dB. Thus, the allowable error in each LSP is approximately 20% to its closest neighbor.

According to listening tests, a 2 dB average spectral error is as big as one can tolerate. Thus the allowable-frequency tolerance of each line spectrum, as obtained from Fig. 12, is approximately 20% of the frequency separation to its closest neighbor.

Just-Noticeable LSP Difference

Because the human ear is insensitive to small differences in frequencies, each LSP has an allowable frequency tolerance (Fig. 13). If two LSP sets have each LSP member fall inside their respective tolerance, then the two LSP sets can be treated as equivalent. This property is to be used later for vector quantization.

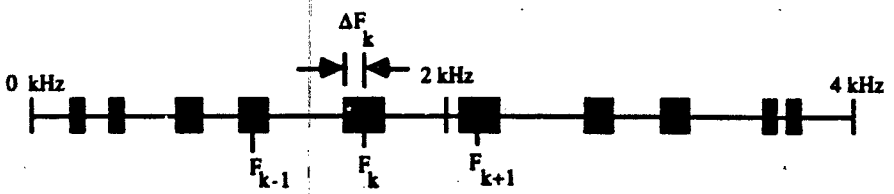


Fig. 13 — Frequency tolerance around each line spectrum. When each line spectrum is disturbed within its tolerance, the synthesized speech sounds no different. F_k is the kth line spectrum arranged in ascending order: $F_1 < F_2 < \ldots < F_k < \ldots < F_{10}$. As shown in Fig. 14, the allowable tolerance of each line spectrum (ΔF_k) is approximately 20, 30, and 40%, for the line spectrum located below 1 kHz, between 1 and 2 kHz, and above 2 kHz. If the LSPs are perturbed by this amount from frame to frame, the resultant speech will not be degraded significantly.

The magnitude of LSP tolerance (shown in Fig. 13) can be established by using the effect of the hearing sensitivity to frequency difference (Fig. 11) and the spectral-error sensitivity of LSP (Fig. 12). The result is plotted in Fig. 14. To verify the validity of this relationship, we synthesized speech while perturbing each line spectrum by the amount defined in Fig. 14. We noticed that synthesized speech contained a just-perceivable amount of flutter.

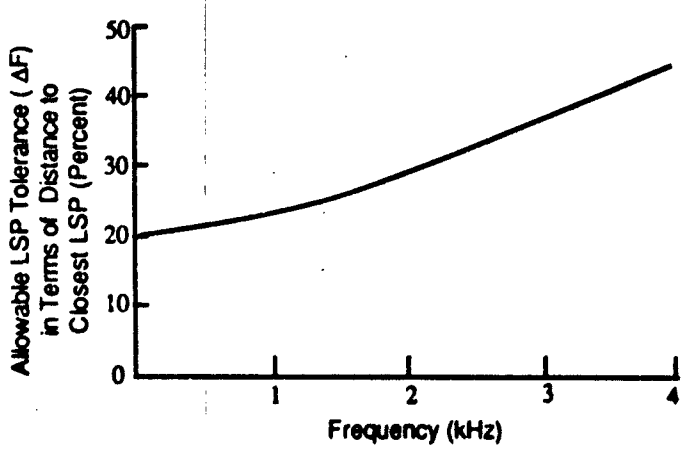


Fig. 14 — Allowable frequency tolerance of each line spectrum based on both ear's sensitivity to frequency differences and the spectral-error sensitivity of the LSP for a 2 dB average. This figure applies to the first through tenth LSP frequencies; therefore, ΔF is free from index k. This relationship becomes vital to vector quantization of LSPs.

Bit Assignments

The single most critical factor for the design of an 800-b/s voice processor is the bit assignments for speech parameters because the total number of bits available to encode speech information is only 16 bits per frame (or 32 bits per two frames), as noted in Table 4. To encode speech parameters efficiently, we take the following new approaches:

- Joint encoding of parameters from two adjacent frames — We transmit two sets of parameters for two frames as a unit, except for the pitch period. By transmitting two frames of data as a unit, we can use the parameter correlation existing in two adjacent frames. For example, we cannot change our speaking volume from the maximum to minimum over one frame of time (20 ms). Hence such a transition can be eliminated from the coding of amplitude information. A similar argument holds for spectral parameters (i.e., LSPs). We discuss more about this later.

- Speech-spectrum-dependent voicing decision — Customarily, voicing information is encoded in one bit. In our approach, the voicing information is embedded in the spectral parameters. For a given LSP set, the voicing decision is predetermined; no voiced speech is without the first formant frequencies. In essence, the presence and absence of the first formant frequency determines the voicing state. To avoid catastrophic error, we designate the voicing decision into one of the 16 possible states: 0 indicates totally voiced, 15 indicates totally unvoiced.

Table 4 — Bit Assignments for 800-b/s Voice Encoding

General Information		
Sampling rate	8 kHz ± 0.1%	
Data rate	800 b/s	
Frame size	20 ms	
Frame rate	50 Hz	
No. of bits per 2 frames	32 bits	
Encoded Parameters Per Two Frames		
Filter and voicing parameters	Line-spectrum pairs (with voicing information)	17 bits
Excitation-signal parameters	Amplitude information	9
	pitch period	5
Other	Synchronization	1
	TOTAL	32 bits per two frames

As usual, a synchronization bit is an alternating 1 and 0 separated by 31 bits. We describe encoders and decoders for other parameters in subsequent sections. How to encode pitch, amplitude information, and LSPs are critical issues in the 800-b/s LPC, and they are also discussed.

Pitch Encoder/Decoder

The pitch period is encoded into one of the 32 steps for pitch periods from 20 to 120 speech sampling intervals (Table 5). The pitch resolution is 12 steps per octave (equi-tempered chromatic scale). As noted in Table 5, the upper limit of the pitch period is 120, which corresponds to the fundamental pitch

frequency of 66.67 Hz. This is not a serious limitation because the average pitch frequency for male voices lies between 100 to 130 Hz, and the male pitch frequency seldom drops below 66.67 Hz.

Table 5 — Pitch Encoding/Decoding Table. Pitch periods of 20 and 120 correspond to the fundamental pitch frequencies of 400 Hz and 66.666 Hz, respectively. As noted, the pitch resolution of the 800-b/s LPC is as good as that of the 2400-b/s except that the low end of pitch range is curtailed.

Pitch Period*	Pitch Code	Decoded Pitch	Pitch Period*	Pitch Code	Decoded Pitch	Pitch Period*	Pitch Code	Decoded Pitch
20	0	20	40	12	40	80	24	80
21	1	21	42	13	42	84	25	85
22	2	22	44	14	44	88	26	90
23	3	23	46	15	47	92	26	90
24	4	24	48	15	47	96	27	95
25	5	26	50	16	50	100	28	101
26	5	26	52	17	53	104	28	101
27	6	28	54	17	53	108	29	107
28	6	28	56	18	57	112	30	113
29	7	30	58	18	57	116	30	113
30	7	30	60	19	60	120	31	120
31	8	32	62	20	63	124	31	120
32	8	32	64	20	63	128	31	120
33	9	34	66	21	67	132	31	120
34	9	34	68	21	67	136	31	120
35	10	36	70	22	71	140	31	120
36	10	36	72	22	71	144	31	120
37	11	38	74	23	75	148	31	120
38	11	38	76	23	75	152	31	120
39	12	40	78	24	80	156	31	120

*Pitch values allowed by the 2400-b/s LPC.

Amplitude Encoder/Decoder (Vector Quantizer)

The amplitude parameter is the root-mean-square value of the speech waveform computed for each frame. We vectorially quantize two consecutive amplitude parameters into one index. In this way, improbable amplitude transitions are eliminated from the coding table to achieve more efficient quantization. To perform vector quantization, we initially quantize the individual amplitude parameter independently into one of 26 amplitude levels listed in Table 6.

Table 6 — Individually Quantized Amplitude Levels from Two Consecutive Frames (A1 and A2) and Amplitude Index

Amplitude Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A1 or A2	0	2	5	8	11	14	17	20	23	28	33	42	51	62	76	93	113	138	168	206	251	307	375	459	561	686

Among 767 (= 26 × 26) possible amplitude transitions, only 512 are significant according to extensive analyses of various speech samples. Table 7 shows the population counts of two amplitudes (A1 and A2) for the amplitude levels specified in Table 6.

Table 7 — Statistics of Amplitude Parameter Transitions over Two Consecutive Frames. This table lists the number of amplitude transitions from one frame to the next (i.e., A1 to A2). As noted, some amplitude transitions doe not occur in actual speech samples. The allowable amplitude transitions are contained in the shaded area. Thus, by vectorially quantizing A1 and A2, we can reduce the number of bits to encode the amplitude parameter.

A1 \ A2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	14308	313	83	60	45	34	31	32	33	41	41	50	34	17	14	15	7	5	3	2	0	0	0	0	0	0
2	589	887	124	45	39	27	18	16	21	26	22	23	23	13	11	9	2	3	1	4	0	0	1	0	0	0
3	114	360	283	81	44	25	20	14	16	23	24	14	19	10	16	6	13	4	1	0	0	0	0	0	0	0
4	45	130	181	156	72	28	19	19	16	14	11	13	17	12	15	8	5	4	1	1	1	0	0	0	0	0
5	33	60	116	130	127	48	28	32	21	24	28	15	19	20	13	19	7	5	2	4	0	0	0	0	0	0
6	19	49	50	88	78	72	53	30	28	23	16	26	15	18	17	7	11	5	1	0	0	0	0	0	0	0
7	18	32	35	61	62	69	46	27	33	16	18	21	19	17	10	14	7	5	2	1	1	0	0	0	0	0
8	1	25	24	37	39	57	52	48	33	35	25	32	26	26	15	16	5	3	5	2	0	2	0	0	0	0
9	8	21	24	28	43	54	52	52	77	51	49	38	26	27	19	16	12	10	5	2	1	0	0	0	0	0
10	9	18	26	22	42	40	51	60	83	99	66	56	33	33	23	25	14	14	4	3	0	0	0	0	0	0
11	4	19	29	26	29	30	32	47	81	115	127	100	62	58	32	32	20	19	11	9	1	0	0	0	0	0
12	3	19	21	15	24	30	29	44	46	89	153	165	110	86	75	45	37	22	13	5	3	0	0	0	0	0
13	1	7	16	19	11	17	29	29	33	64	94	191	203	127	106	69	26	21	15	7	8	2	0	0	0	0
14	0	9	11	10	11	14	16	13	38	40	67	140	199	291	162	90	69	38	16	9	6	3	3	0	0	0
15	1	4	7	5	16	9	12	13	18	27	58	66	127	264	282	196	112	65	17	17	11	5	0	0	0	0
16	0	3	4	7	10	11	13	14	19	24	42	53	79	114	270	344	222	121	68	30	10	6	1	0	0	0
17	1	1	2	4	4	6	6	8	10	14	28	36	58	75	138	298	386	182	90	44	24	11	3	0	0	0
18	0	0	1	5	2	3	2	5	4	4	12	19	32	33	53	119	283	436	205	76	26	12	4	0	2	0
19	0	0	1	2	5	2	0	2	2	5	9	12	16	18	29	47	96	295	428	181	60	14	14	1	0	0
20	0	0	0	1	0	2	2	1	3	2	5	2	5	11	10	24	39	64	263	367	139	32	6	2	0	0
21	0	0	0	1	0	0	0	0	2	0	2	3	2	6	2	11	8	31	65	196	341	112	22	4	0	0
22	0	0	0	0	0	0	0	0	0	0	1	2	2	1	4	6	4	8	16	29	124	231	55	4	2	0
23	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	2	3	10	16	78	124	28	1	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	2	4	22	45	7	3
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	3	3	8	2
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	2	3

Good amplitude resolution is highly critical to speech intelligibility. By performing vector quantization we can achieve an amplitude quantization at 4.5 bits per frame, which is nearly as good as the five-bit amplitude quantization of the typical 2400-b/s LPC. A saving of a half bit per frame is significant to the implementation of an 800-b/s processor because the total number of bits per frame is only 16. Table 8 is a vector quantization table for two sets of amplitude parameters.

LSP Encoder/Decoder (Matrix Quantizer)

Encoding filter coefficients is critical to the overall speech quality and intelligibility. As stated previously, the 2400-b/s LPC uses 41 bits to encode 10 filter coefficients for each frame, where we have only 17 bits to encode LSPs over two frames (see Table 1). Therefore, much of our research effort has been concentrated on efficient encoding of the filter coefficients.

Previously, pattern matching (often called vector quantization) of filter coefficients has shown remarkable results [9, 15, 16]. In this approach, speech is synthesized from the filter coefficients selected from the reference templates that are free from nonspeech sounds. We again use a similar technique but

Table 8 — Coding/Decoding Table for Two Amplitude Parameters (A1 and A2)

A1	A2	Code
1	1	0
	2	1
	3	2
	4	3
	5	4
	6	5
	7	6
	8	7
	9	8
	10	9
	11	10
	12	11
	13	12
	14	13
	15	14
	16	15
	17	16
	18	17
	19	18
	20	19
2	1	20
	2	21
	3	22
	4	23
	5	24
	6	25
	7	26
	8	27
	9	28
	10	29
	11	30
	12	31
	13	32
	14	33
	15	34
	16	35
	17	36
	18	37
	19	38
	20	39
	21	40

A1	A2	Code
3	1	41
	2	42
	3	43
	4	44
	5	45
	6	46
	7	47
	8	48
	9	49
	10	50
	11	51
	12	52
	13	53
	14	54
	15	55
	16	56
	17	57
	18	58
	19	59
	20	60
	21	61
4	1	62
	2	63
	3	64
	4	65
	5	66
	6	67
	7	68
	8	69
	9	70
	10	71
	11	72
	12	73
	13	74
	14	75
	15	76
	16	77
	17	78
	18	79
	19	80
	20	81
	21	82
	22	83

A1	A2	Code
5	1	84
	2	85
	3	86
	4	87
	5	88
	6	89
	7	90
	8	91
	9	92
	10	93
	11	94
	12	95
	13	96
	14	97
	15	98
	16	99
	17	100
	18	101
	19	102
	20	103
	21	104
	22	105
6	1	106
	2	107
	3	108
	4	109
	5	110
	6	111
	7	112
	8	113
	9	114
	10	115
	11	116
	12	117
	13	118
	14	119
	15	120
	16	121
	17	122
	18	123
	19	124
	20	125
	21	126
	22	127

A1	A2	Code
7	1	128
	2	129
	3	130
	4	131
	5	132
	6	133
	7	134
	8	135
	9	136
	10	137
	11	138
	12	139
	13	140
	14	141
	15	142
	16	143
	17	144
	18	145
	19	146
	20	147
	21	148
	22	149
8	1	150
	2	151
	3	152
	4	153
	5	154
	6	155
	7	156
	8	157
	9	158
	10	159
	11	160
	12	161
	13	162
	14	163
	15	164
	16	165
	17	166
	18	167
	19	168
	20	169
	21	170
	22	171

A1	A2	Code
9	1	172
	2	173
	3	174
	4	175
	5	176
	6	177
	7	178
	8	179
	9	180
	10	181
	11	182
	12	183
	13	184
	14	185
	15	186
	16	187
	17	188
	18	189
	19	190
	20	191
	21	192
	22	193
10	1	194
	2	195
	3	196
	4	197
	5	198
	6	199
	7	200
	8	201
	9	202
	10	203
	11	204
	12	205
	13	206
	14	207
	15	208
	16	209
	17	210
	18	211
	19	212
	20	213
	21	214
	22	215

A1	A2	Code
11	1	216
	2	217
	3	218
	4	219
	5	220
	6	221
	7	222
	8	223
	9	224
	10	225
	11	226
	12	227
	13	228
	14	229
	15	230
	16	231
	17	232
	18	233
	19	234
	20	235
	21	236
	22	237
12	1	238
	2	239
	3	240
	4	241
	5	242
	6	243
	7	244
	8	245
	9	246
	10	247
	11	248
	12	249
	13	250
	14	251
	15	252
	16	253
	17	254
	18	255
	19	256
	20	257
	21	258
	22	259

Table 8 (Cont'd) — Coding/Decoding Table for Two Amplitude Parameters (A1 and A2)

A1	A2	Code
13	1	260
	2	261
	3	262
	4	263
	5	264
	6	265
	7	266
	8	267
	9	268
	10	269
	11	270
	12	271
	13	272
	14	273
	15	274
	16	275
	17	276
	18	277
	19	278
	20	279
	21	280
	22	281
	23	282
14	1	283
	2	284
	3	285
	4	286
	5	287
	6	288
	7	289
	8	290
	9	291
	10	292
	11	293
	12	294
	13	295
	14	296
	15	297
	16	298
	17	299
	18	300
	19	301
	20	302
	21	303
	22	304
	23	305

A1	A2	Code
15	1	306
	2	307
	3	308
	4	309
	5	310
	6	311
	7	312
	8	313
	9	314
	10	315
	11	316
	12	317
	13	318
	14	319
	15	320
	16	321
	17	322
	18	323
	19	324
	20	325
	21	326
	22	327
	23	328
16	1	329
	2	330
	3	331
	4	332
	5	333
	6	334
	7	335
	8	336
	9	337
	10	338
	11	339
	12	340
	13	341
	14	342
	15	343
	16	344
	17	345
	18	346
	19	347
	20	348
	21	349
	22	350
	23	351

A1	A2	Code
17	1	352
	2	353
	3	354
	4	355
	5	356
	6	357
	7	358
	8	359
	9	360
	10	361
	11	362
	12	363
	13	364
	14	365
	15	366
	16	367
	17	368
	18	369
	19	370
	20	371
	21	372
	22	373
	23	374
18	2	375
	3	376
	4	377
	5	378
	6	379
	7	380
	8	381
	9	382
	10	383
	11	384
	12	385
	13	386
	14	387
	15	388
	16	389
	17	390
	18	391
	19	392
	20	393
	21	394
	22	395
	23	396
	24	397
	25	398

A1	A2	Code
19	3	399
	4	400
	5	401
	6	402
	7	403
	8	404
	9	405
	10	406
	11	407
	12	408
	13	409
	14	410
	15	411
	16	412
	17	413
	18	414
	19	415
	20	416
	21	417
	22	418
	23	419
	24	420
	25	421
20	4	422
	5	423
	6	424
	7	425
	8	426
	9	427
	10	428
	11	429
	12	430
	13	431
	14	432
	15	433
	16	434
	17	435
	18	436
	19	437
	20	438
	21	439
	22	440
	23	441
	24	442
	25	443

A1	A2	Code
21	4	444
	5	445
	6	446
	7	447
	8	448
	9	449
	10	450
	11	451
	12	452
	13	453
	14	454
	15	455
	16	456
	17	457
	18	458
	19	459
	20	460
	21	461
	22	462
	23	463
	24	464
	25	465
22	11	466
	12	467
	13	468
	14	469
	15	470
	16	471
	17	472
	18	473
	19	474
	20	475
	21	476
	22	477
	23	478
	24	479
	25	480

A1	A2	Code
23	11	481
	12	482
	13	483
	14	484
	15	485
	16	486
	17	487
	18	488
	19	489
	20	490
	21	491
	22	492
	23	493
	24	494
	25	495
24	19	496
	20	497
	21	498
	22	499
	23	500
	24	501
	25	502
	26	503
25	22	504
	23	505
	24	506
	25	507
	26	508
26	24	509
	25	510
	26	511

take it one step further. We apply a pattern matching technique for jointly encoding filter coefficients from two adjacent frames. In this way, we not only eliminate nonspeech sounds from encoding, but we also eliminate improbable filter coefficient transitions across two adjacent frames. In essence, we perform two-dimensional vector quantization (matrix quantization). The basic method of matrix quantization is similar to vector quantization except that we jointly quantize 20 line-spectral frequencies (10 from each frame).

LSP Template Collection

We generate a representative number of LSP templates by analyzing many representative voice samples. LSP templates are generated by the following steps:

Step 1: The first incoming LSP matrix (two LSP vectors from two consecutive frames) is the first LSP template, and it is stored in memory.

Step 2: The second incoming matrix is compared with the stored template. If all the incoming LSPs fall into the tolerance of the respective LSP members of the template, this incoming LSP matrix is regarded as being the same family, and therefore it will be discarded. Otherwise, it will be stored as a new template.

Step 3: Step 2 is repeated until the maximum allowable template size (i.e., $2^{17} = 131{,}072$) is reached. Actually we collect more than the maximum number, pending elimination of least-frequently-used templates later on to meet the required maximum template size.

A similar template collection approach has been used in our previous 800-b/s voice processor that achieved a DRT score of 87 [9]. Likewise a similar approach was also successfully used by Gold [16] for the channel vocoder, and Paul [17] for the spectral-envelope-estimation vocoder. We did not consider updating speaker's templates during communication because it is not a viable approach for the tactical voice terminal where the average duration of tactical voice communication is on the order of a few seconds.

The intelligibility of synthesized speech will be low if the reference templates lack a variety of voice characteristics. If so, new speaker's parameters will be far outside of the hyperspace defined by the templates. Therefore, the resultant speech quality will be poor. No speech improvement is expected by clustering or reclustering templates. What is desirable is to spread out the parameter space as much as feasible by introducing distinctly different voice parameters during template collection.

LSP Template Storage in Tree Arrangement

An exhaustive search of 131,072 LSP templates in two frames cannot be performed in real time with present-day hardware. Thus, the templates must be partitioned in such a way that only a fraction of the total templates are searched. We present a method of LSP template partitioning where the maximum number of templates in any one group is only 2048.

(A) Initial Partitioning

Since each LSP template has two voicing decisions associated with it, we initially partition LSP templates into five cases based on the voicing transition over the two frames. We use a 16-level voicing decision with a range from 0 to 15: 0 and 15 imply totally voiced and totally unvoiced, respectively.

Case 1. Totally unvoiced to totally unvoiced ($v1=v2=15$): This case includes fricatives, plosives, and silence. The number of templates is 1024, which can be searched exhaustively.

Case 2. Both frames are partially voiced ($15 \geq v1 > 0$ and $15 \geq v2 > 0$): This case is divided into four groups (each having 2048 templates) based on the voicing decision levels (Fig. 15). The 2048 templates in each group can be searched exhaustively.

Case 3: The first frame is totally voiced and the second frame is not totally voiced ($v1=0$ and $v2 \neq 0$): This case is for the trailing end of words or phrases. The template size is 2048, which can be searched exhaustively.

Case 4: The first frame is not totally voiced but the second frame is totally voiced ($v1 \neq 0$ and $v2=0$): This case is for speech onsets and is critical to speech intelligibility. There are 16,384 LSP templates included here that need further partitioning.

Case 5: Both frames are totally voiced ($v1=0$ and $v2=0$): This case is for vowels. There are 103,424 LSP templates here that will require further partitioning.

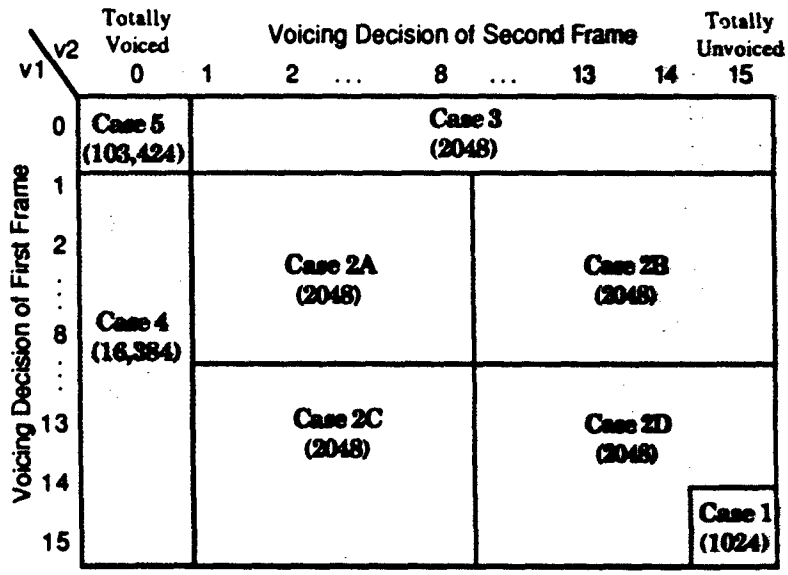


Fig. 15 — The first-stage LSP template partitioning based on voicing transitions. The number of templates in each case is given inside the bracket. These figures are based on speech samples of 420 speakers uttering 8 sentences each, excerpted from the Texas Instrument - Massachusetts Institute of Technology (TIMIT) Acoustic-Phonetic Speech Data Base [18]. The LSP templates for cases 1, 3, and 5 (boxes with lighter shade) can be searched exhaustively, but the LSP templates for cases 2 and 4 (boxes with darker shade) must be partitioned further.

(B) Further Partitioning Based on Closely Spaced Line-Spectral Frequencies

We have 16,384 LSP templates for Case 4 and 103,424 LSP templates for Case 5. They must be further partitioned. These LSP templates represent voiced speech (vowels) where resonant frequencies are critical to speech intelligibility. We group LSP templates of similar spectral characteristics. In other words, LSP templates obtained from, for example, /i/ will not be grouped with LSP templates obtained

from /u/. Template grouping in terms of similar spectral characteristics can be exploited to improve tolerance to bit errors because an error in the least significant bit will result in a template with a similar sound. To achieve our objective, we define the index of line-spectral frequency separation:

- Let line-spectral frequencies be denoted by $f_1, f_2, \ldots, f_{10}$ where $f_1 < f_2 < , \ldots , < f_{10}$, as illustrated in Fig. 16.
- Note that the frequency separation between f_1 and f_2 does not fluctuate greatly within the voiced region. Thus, we will not incorporate f_1 and f_2 in the LSP template partitioning.
- Similarly, the frequency separation between f_9 and f_{10} does not fluctuate significantly. Thus, the separation between f_9 and f_{10} will not be exploited in the LSP template partitioning.
- If the frequency separations between f_1 and f_2 and between f_9 and f_{10} are not considered, there are seven possible frequency separations remaining, as indicated in Fig. 16. The ith frequency separation is defined as

$$\Delta f_i = f_{i+2} - f_{i+1} \quad i = 1,2,\ldots, 7.$$

The index corresponding to the smallest Δf_i is dependent on the vowel (see Fig. 1b for example). We will use as many as four sets of closely spaced frequency separations to partition LSP templates for Case 5.

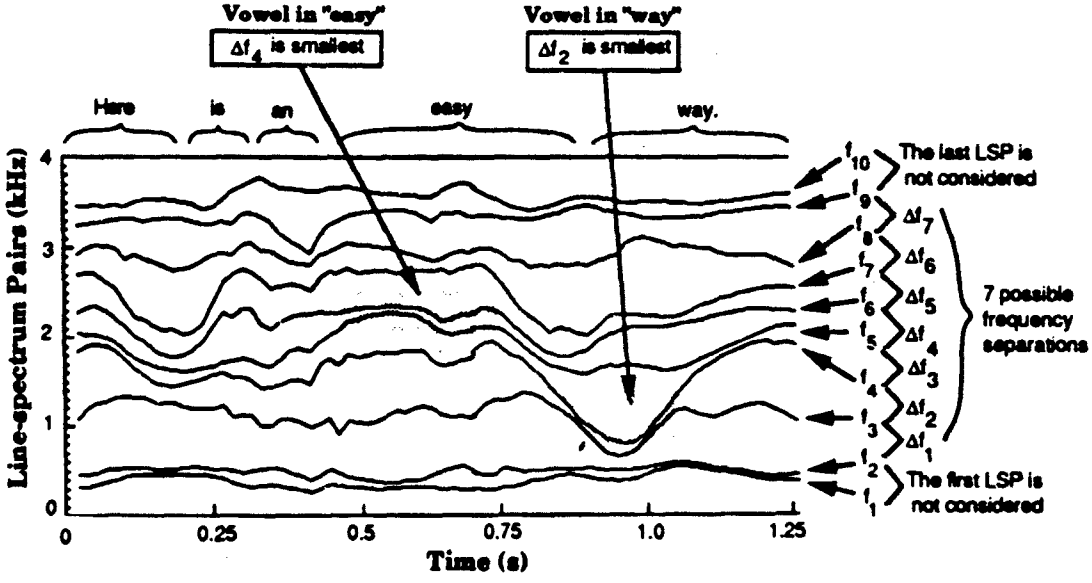


Fig. 16 — The LSP trajectories for "Here is an easy way." As noted, the first and last LSPs are not very sensitive to speech content. Therefore, we will not use these two LSPs for template partitioning. As illustrated earlier in Fig. 10, closely-spaced line-spectrum frequencies are located near speech resonant frequencies. Since each line-spectrum frequency is distributed over a limited frequency range [9], the indices of the three or four closest line-spectrum frequency separations characterize vowels adequately for template partitioning.

Further LSP Template Partitioning for Case 4

The voicing transition is from partially voiced to totally voiced ($v1 \neq 0$ and $v2 = 0$). The total number of LSP templates is 16,384 (Fig. 15). Since only the second frame is voiced, we use the indices of the two closely spaced line-spectral frequencies of the second frame to partition LSP templates. Figure 17 shows LSP templates stored in a tree arrangement for Case 4.

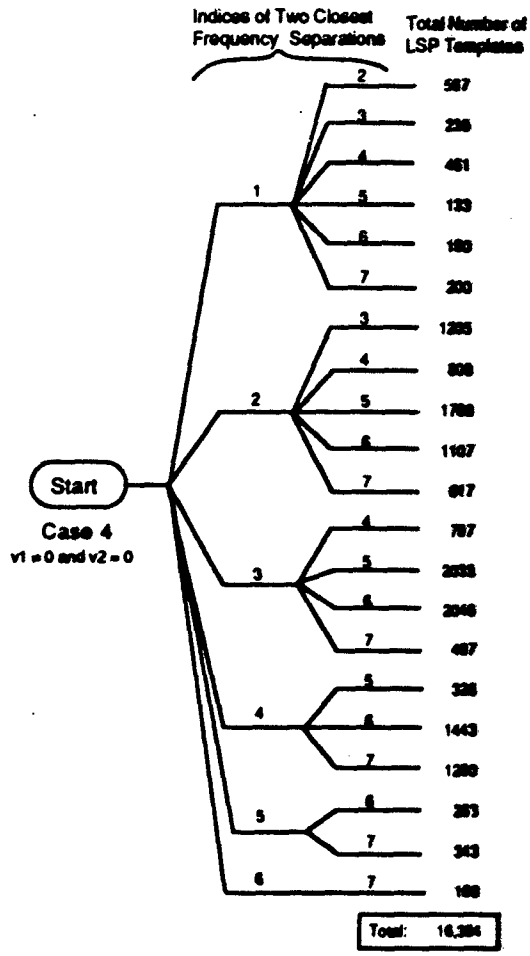


Fig. 17 — LSP partition for Case 4 where the first frame is not totally voiced, but the second frame is totally voiced ($v1 \neq 0$ and $v2 = 0$). There are 21 possible combinations for choosing two out of seven frequency separations. The partition size for each of the 21 possible groups is listed in the right-hand column. In one group, the template size reached 2172 which was clamped to 2048.

Further LSP Template Partitioning for Case 5

Case 5 is where the voicing decisions for both frames are totally voiced ($v1=v2=0$). Thus, Case 5 represents vowels in both frames. If the speech is a sustained vowel over the two frames, the indices of the closely spaced frequency separations will be identical in both frames. For transitional vowels, they are expected to be different. According to our analysis data, the number of templates from sustained vowels is approximately one order of magnitude greater than the number of transitional vowels. Since there are more sustained vowels, we will successively sort out sustained vowels based on the degree of stationarity.

Figure 18 is a tree diagram of further partitioning of LSP templates for Case 5. Initially, we separate LSP templates for the cases where indices of the four closest frequency separations are identical in both frames. We repeat a similar partitioning by using three and two indices. The LSP templates that failed the above three sequential tests are probably transitional vowels. They will be partitioned into a two-dimensional matrix made of 7×7 elements by using the index of the minimum frequency separation from each frame. Note that in this final sorting, the index of the minimum frequency separation from frame 1 may be different from that of frame 2.

LSP Template Matching

The incoming LSP matrix (LSP sets from two adjacent frames) are compared with all of the LSP templates (each template is likewise made of two LSP sets). The index corresponding to the closest match is transmitted. We use the error criterion expressed as the sum of the absolute values of weighted differences between two sets of LSP matrices, $\{F_a\}$ and $\{F_b\}$, each composed of 20 line-spectrum frequencies. Thus,

$$d(F_a, F_b) = \sum_{i=1}^{20} |w_a(i)\,[F_a(i) - F_b(i)]| \tag{26}$$

and

$$d(F_b, F_a) = \sum_{i=1}^{20} |w_b(i)\,[F_a(i) - F_b(i)]| \tag{27}$$

where $w_a(i)$ and $w_b(i)$ are the weights of the ith line spectrum of $\{F_a\}$ and $\{F_b\}$, respectively. The magnitude of the weighting factor is inversely proportional to the LSP tolerance (ΔF) (i.e., closely spaced and low-frequency line spectra are more heavily weighted). For each comparison of two LSP matrices, we generate two-way errors based on both Eqs. (24) and (25); then we choose the largest error of the two. We compute the weighting factors beforehand and store them with the LSP templates.

5. INTELLIGIBILITY TEST SCORES

The Diagnostic Rhyme Test (DRT) evaluates the discriminability of initial consonants of monosyllable rhyming word pairs. For many years, DRT scores have been widely used as a diagnostic tool to refine voice processors. Likewise, it has been effectively used to rank several competing voice processors. Over the years, an extensive amount of DRT data has been collected from different voice processors under varied operating conditions. According to our experience, DRT scores are dependable (i.e., scores are repeatable under retesting), and they often reveal latent defects of synthetic speech that are not easily discernible through casual listening.

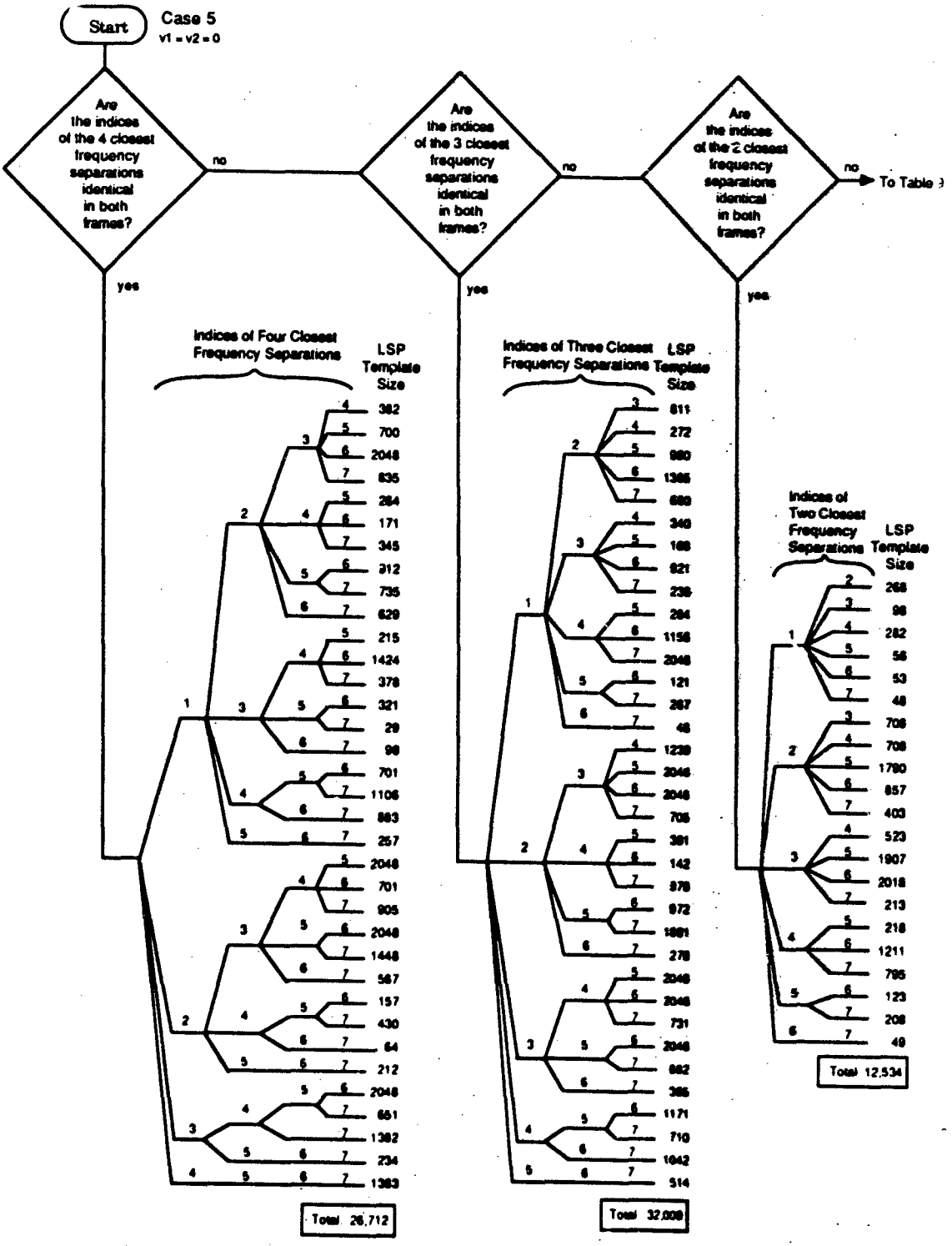


Fig. 18 — Partitioning of LSP templates when both frames are totally voiced (Case 5)

Table 9 — Final LSP Template Partitioning of Case 2. LSP templates which failed three successive tests (see Fig. 18) are grouped based on the index on the minimum frequency separation. The figures are the total number of LSP templates in each group. When the numbers exceeded 2048, they were clamped to 2048. The total number of LSP templates is 32,169.

		Index of Minimum Frequency Separation (Frame #2)						
		1	2	3	4	5	6	7
Index of Minimum Frequency Separation (Frame #1)	1	565	358	219	359	139	160	152
	2	222	2048	1687	397	674	578	315
	3	197	1175	2048	1333	1007	1434	541
	4	282	253	845	2048	609	838	636
	5	114	447	638	463	1447	412	252
	6	172	378	888	516	438	2048	283
	7	138	218	344	426	218	266	944

If speech is severely degraded, however, additional tests may be needed because speech with poor DRT scores (i.e., below 70) can still be functional if the contextual information is limited. If the listener understands the topic of conversation, operating environment, nature of mission, etc., he (or she) can anticipate or predict the message; thus, communication may be feasible even if the intelligibility of the voice system is rather low. In this case, word discrimination tests may be more meaningful than initial consonant discrimination tests such as DRT. We tested both for our 800-b/s voice processor.

Diagnostic Rhyme Test

Based on the 800-b/s voice processor described in the preceding sections, we ran several DRT tapes (Table 10). Three male speakers (CH, LL, and RH) are used for this test. As far as we can determine, these are the highest DRT scores for any 800-b/s voice processor. For comparison, DRT scores for the latest 2400-b/s LPC (LPC-10e) are also entered in this table. Run 1 had a one-way error criterion; Run 2 used a two-way error criterion expressed in Eqs. (24) and (25); and Run 3 performed a tree search.

We can summarize a few significant points from these intelligibility scores:

- The 800-b/s voice algorithm consistently scored 92 when using the DRT under slightly different test conditions. Since we have performed and scored over a time period of several months, the stability of the algorithm performance is remarkable.

- The strength of the 800-b/s algorithm lies in the attribute *sibilation*. The algorithm discriminated the following word pairs more successfully than the 2400-b/s LPC:

ZEE - THEE	JILT - GILT	JEST - GUEST
CHEEP - KEEP	SING - THING	CHAIR - CARE

Table 10 — DRT Scores of the 800-b/s Voice Processor

DRT Attribute		800 b/s			2400 b/s (LPC-10e)
		#1	#2	#3	
Voicing	Distinguishes /b/ from /p/, /d/ from /t/, /v/ from /f/, etc.	96.9	97.4	95.1	95.1
Nasality	Distinguishes /n/ from /d/, /m/ from /b/, etc.	96.1	95.1	96.9	96.9
Sustention	Distinguishes /f/ from /p/, /b/ from /v/, /t/ from /θ/, etc.	86.7	87.5	82.8	88.3
Sibilation	Distinguishes /s/ from /θ/, /ʃ/ from /d/, etc.	96.4	98.2	95.6	93.8
Graveness	Distinguishes /p/ from /t/, /b/ from /d/, /m/ from /n/, etc.	81.5	80.5	79.9	87.0
Compactness	Distinguishes /g/ from /d/, /k/ from /t/, /ʃ/ from /s/, etc.	95.1	95.3	97.1	96.4
	TOTAL	92.1	92.3	91.2	92.9

- The weakness of the 800-b/s algorithm lies with the attribute *graveness*. The following word-pairs were difficult for the algorithm:

PEEK - TEAK	BID - DID	BANK - DANK
FAD - THAD	WAD - ROD	MOON - NOON
YIELD - WIELD	GILL - DILL	KEY - TEA
HIT - FIT	KEG - PEG	SHOW - SO

- In our 800-b/s voice processor, the voicing decision was attached to each LSP template. In other words, for a given spectral envelope, the voicing decision is predetermined. Although for some cases this may not be a good procedure, this is an approach that should be studied more.

- For the past 10 years, intelligibility of 800-b/s voice processors has improved 10 points (Fig. 19). The improvement of intelligibility is in part contributed by the availability of powerful signal processors in recent years. Now we can perform pattern matching with the number of templates in the several thousands.

ICAO Phonetic Alphabet Word Test

Recently, Astrid Schmidt-Nielsen of NRL made a study to provide a better understanding of the effects of very degraded speech on human communication performance [19]. In particular, she related DRT scores to the discrimination scores of the International Civil Aviation Organization (ICAO) phonetic alphabet words (ALPHA, BRAVO, CHARLIE, etc.). She noted that the word intelligibility based on

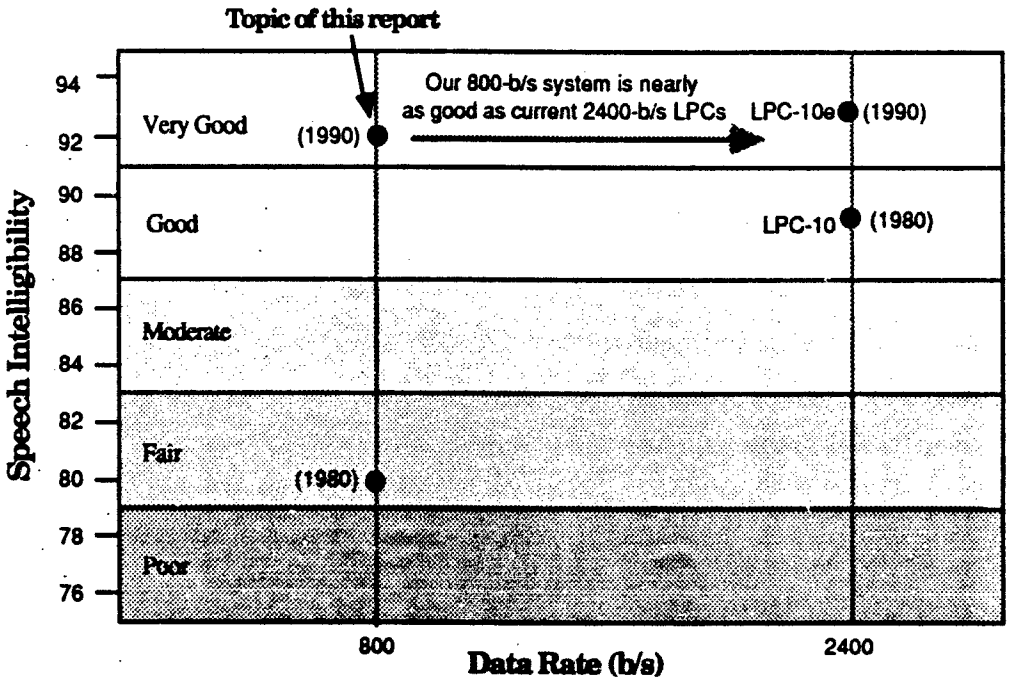


Fig. 19 — DRT improvements in the 800 and 2400-b/s voice-processing algorithms over the past 10 years. This chart demonstrates that long-term research can steadily improve speech intelligibility. Now the intelligibility of an 800-b/s voice processor can be called "very good."

a distinctive vocabulary like the ICAO phonetic alphabet remains rather high even when DRT scores fall into the poor range.

We used the source tape consisting of two male and two female speakers, each uttering 26 ICAO phonetic alphabet words and the names of the first ten digits (zero to niner), which are repeated in three different randomized sequences. Thus, the total number of word pairs in the source tape is ($4 \times 36 \times 3 = 432$ words). Similar to the evaluation of DRT scores, the ICAO phonetic word test scores are evaluated by a third party who is not associated with the authors' voice processor development. The scores are plotted in Fig. 20.

6. CONCLUSIONS

After nearly a decade of research and development, we were able to generate 800-b/s speech that can be classified as "very good" speech. Speech intelligibility of our 800-b/s voice processor exceeds that of the 2400-b/s LPC of a few years ago (viz., ANDVTs that are being widely deployed to support tactical voice communication).

The factors that most contributed to the high intelligibility are: choice of a 20-ms frame, vector quantization of two sets of amplitude parameters, and matrix quantization of two sets of LSP vectors.

We expect that very-low-data-rate voice processors will be increasingly used to enhance bit-error performance, low-probability of intercept, and narrowband voice/data integration.

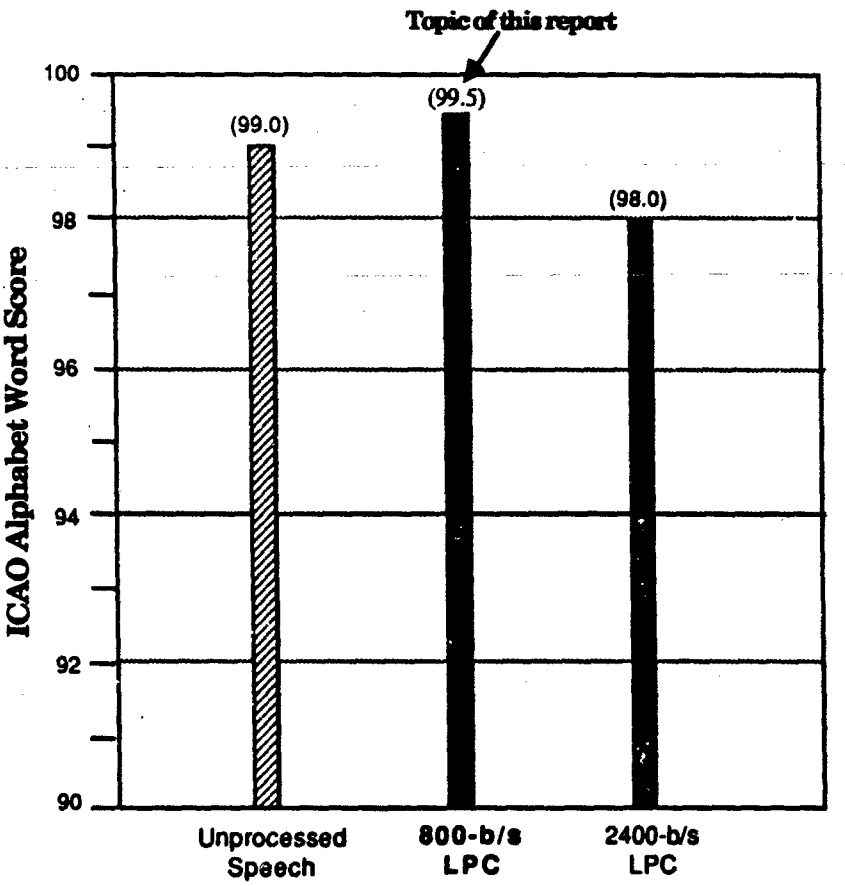


Fig. 20 — ICAO phonetic alphabet word score for the 800-b/s LPC discussed in this report. For reference, similar scores of unprocessed speech and an earlier 2400-b/s LPC are also plotted for reference; they were collected by Schmidt-Nielsen [19], used by permission. This figure implies that the users of our 800-b/s voice processor probably recognize all the ICAO words in benign operating environments.

7. ACKNOWLEDGMENTS

We thank Timothy McChesney and Sharon James of SPAWAR PMW151 for support of this R&D effort. Without their continued support in the past, we could not have written this report.

8. REFERENCES

1. G.S. Kang, "Error-Resistant Narrowband Voice Encoder," NRL Report 9018, Dec. 1986.

2. G.S. Kang, "Narrowband Integrated Voice/Data System Based on the 2400-b/s LPC," NRL Report 8942, Dec. 1985.

3. G.S. Kang and D.C. Coulter, "600 bps voice digitizer," 1976 IEEE ICASSP Record, pp. 91-94, 1976.

4. G.S. Kang and D.C. Coulter, "600-Bits-Per-Second Voice Digitizer," NRL Report 8043, Nov. 1976.

5. D.Y. Wong, B.H. Juang, and A.H. Gray, Jr., "An 800 bits/s Vector Quatization LPC Vocoder," *IEEE Trans. on Acoustics, Speech and Signal Processing* ASSP-30(5), 770-780 (1982).

6. T.E. Carter, D.M. Dlugos, and D.C. LeDoux, "An 800 BPS Real-Time Voice Coding System Based on Efficient Encoding Techniques," IEEE ICASSP Record, pp. 602-605, 1982.

7. L.J. Fransen, "2400- to 800-b/s LPC Rate Converter," NRL Report 8716, June 1983.

8. L.J. Fransen, "Technical Evaluation of Low Data Rate Experimental Terminal (LDRET)," NRL Internal Technical Memorandum prepared for SPAWAR PMW-151, ser: 7520-177A, June 5, 1985.

9. G.S. Kang and L.J. Fransen, "Low-Bit Rate Speech Encoders Based on Line-Spectrum Frequencies (LSFs)," NRL Report 8857, Jan. 1985.

10. G.S. Kang and L.J. Fransen, "Applications of Line-Spectrum Pairs to Low-Bit-Rate Speech Encoders," IEEE ICASSP Record, pp. 244-247, 1985.

11. G.S. Kang and S.S. Everett, "Improvement of the Excitation Source in the Narrow-Band Linear Predictive Vocoder," *IEEE Trans. Acoustics, Speech and Signal Proc.* ASSP-33(2), 377-386 (1985).

12. Federal Standard 1015, "Analog to Digital Conversion of Voice by 2,400 bits/s Linear Predictive Coding," published by General Services Administration (GSA), November 28, 1984. Copies are for the sale at the GSA Specification Unit (WFSIS), Room 6039, 7th and D Street SW, Washington, DC 20407.

13. A.W.F. Huggins, R. Viswanathan, and J. Makhoul, "Quality Rating of LPC Vocoders; Effects of Number of Poles, Quantization and Frame Rate," 1977 IEEE ICASSP Record, pp. 413-416, 1977.

14. P. Kabal and R.P. Ramachandran, "The Computation of Line Spectral Frequencies Using Chebyshev Polynomials," *IEEE Trans. Acoustics, Speech and Signal Proc.* ASSP-34(6) 1419-1426 (1986).

15. P. Ladefoged, *Elements of Acoustic Phonetics* (The University of Chicago Press, Chicago and London, 1974).

16. B. Gold, "Experiments with a Pattern-Matching Channel Vocoder," IEEE ICASSP Record, pp. 32-34, 1981.

17. D.B. Paul, "An 800-b/s Adaptive Vector Quantization Vocoder Using Perceptual Distance Measures," IEEE ICASSP Record, pp. 73-76, 1983.

18. J.S. Carofolo, "DARPA TIMIT Acoustic-Phonetic Speech Database," National Institute of Standards and Technology, Gaithersburg, MD 20899.

19. A. Schmidt-Nielsen, "The Effect of Narrow-Band Digital Processing and Bit Error Rate on the Intelligibility of ICAO Spelling Alphabet Words," *IEEE Trans. Acoustics, Speech and Signal Proc.* ASSP-35(8) 1107-1115 (1987).